Machine Learning: Data Science

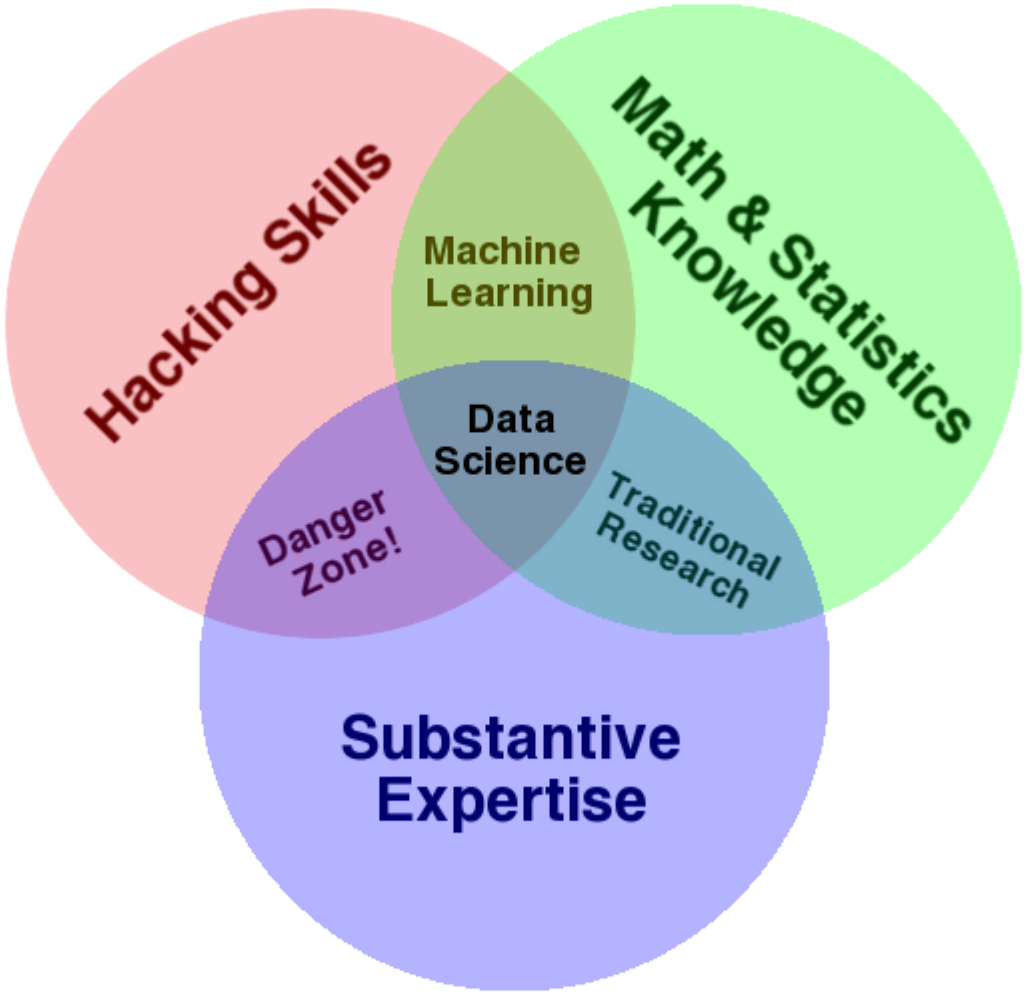
CPSC 501: Advanced Programming Techniques
Winter 2025

Jonathan Hudson, Ph.D
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

Thursday, February 13, 2025

UNIVERSITY OF CALGARY

What is Data Science?



Drew Conway's September 2010 Data Science Venn Diagram
http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Python

- **Python** 3 – Most popular Data Science programming language
  - **R** is popular from the Statistics side of things, **Julia** in Mathematics

1. Very strong support of packages, tutorials, and knowledge
2. Ease of integrating more efficient languages behind scenes like **C++**, etc.
3. Often prototype in **R**/others but implement in **Python** for final production

1. Interpreted, so can be slow (unless break out to **C++** like with ***numpy***)
2. Not built for multi-thread concurrency without effort (unless break out like with ***tensorflow*** for neural networks to use multi-core/GPU)

Packages

Packages - Interaction

IPython (2001)

- https://ipython.org/
- This is a interactive version of **Python**. Differences from regular interactive is that it can store state for reference using In and Out blocks (stored as arrays), and you can also break out of it to run shell commands from inside (like to install library or make folders and manage files)

jupyter notebooks (2014)

- https://jupyter.org/
- GUI to **IPython**,. Your computer hosts a execution kernel and you access it using a web browser web-based interface (like the **Google Colaboratory** and **Syzygy** options given earlier) Can do other languages as well.

Packages – Standard Data Science

- ***numpy*** – manipulation of homogenous array-based data, container
  - https://numpy.org/

- ***pandas*** – 2010, heterogenous and labeled data like tables or databases, brings spreadsheets like functionality to data handling (on top of ***numpy***)
  - https://pandas.pydata.org/

- ***matplotlib*** – publication quality visuals
  - https://matplotlib.org/

- ***seaborn*** – better charts on top of ***matplotlib*** (compete with **R**)
  - https://seaborn.pydata.org/

Packages – Machine Learning

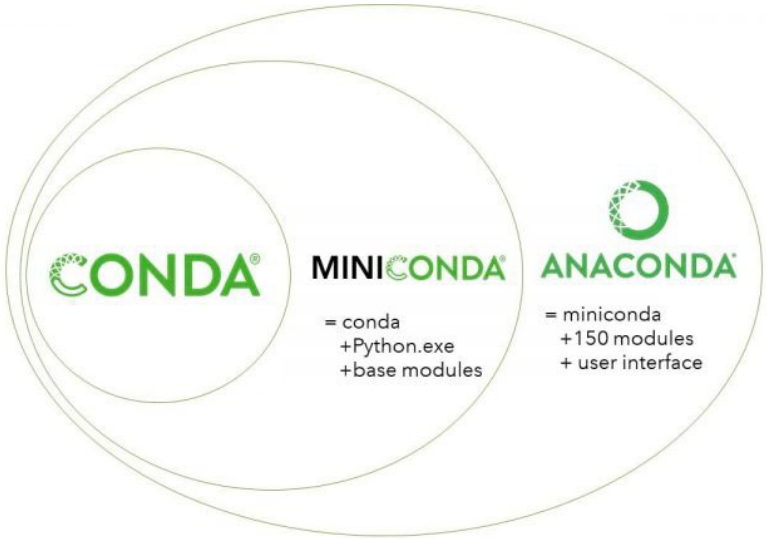
- ***scipy*** – ***scikit*** uses for algorithms, good at storing sparse, common scientific computing tasks (integration, linear algebra, optimization, signal processing, statistics distributions/variables/etc.)
  - https://scipy.org/
- ***scikit-learn*** – 2010 machine learning, classification, regression, clustering, etc.
  - https://scikit-learn.org/stable/index.html
- ***tensorflow*** – 2015 open-sourced distributed neural network library, 2019 v2.0
  - https://www.tensorflow.org/
- ***keras*** – 2015 deep learning library (often on top of ***tensorflow***)
  - https://keras.io/
- ***statmodels*** – classical statistics and econometrics
  - https://www.statsmodels.org/stable/index.html

Installation

Data Science Installation Method (managed)

- **Anaconda** – Distribution of **Python** 3 and package manager that allows easy access to most popular data science libraries
- **Miniconda** – Lighter weight than **Anaconda** as it doesn't pre-download as many packages
  - Generally early learning path is to get miniconda
  - https://docs.conda.io/en/latest/miniconda.html
  - Then install

```
conda install numpy pandas matplotlib seaborn ipython jupyterlab
conda install scipy scikit-learn tensorflow keras statsmodels
```



The first is for Data Science tasks while the second line of installation is for Machine Learning

Data Science Installation (lightweight)

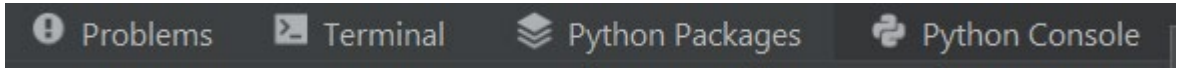
- Could just use available **Jupyter** notebook ***ipython***-like environments such as

1. **Google Colaboratory** https://colab.research.google.com/ (public)
2. **Syzygy** https://ucalgary.syzygy.ca/ (university member host)

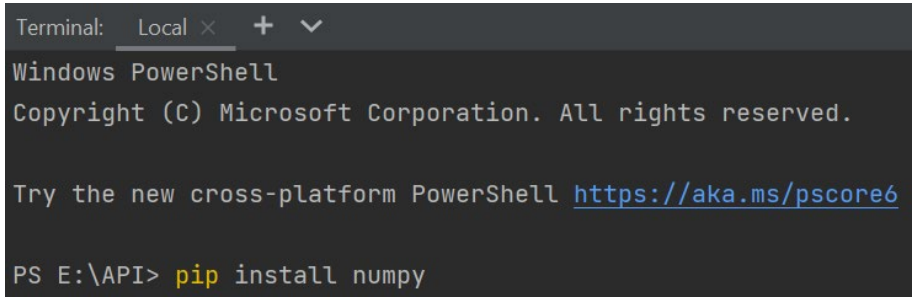
- Or setup your local install of **Python**
  - IDE choice up to you (for Python prefer I like **Pycharm**)
  - Recommend virtual environment (good for UofC lab machines)
  - **pip** for packages (python package manager)

Data Science Installation (IDE-Pycharm)

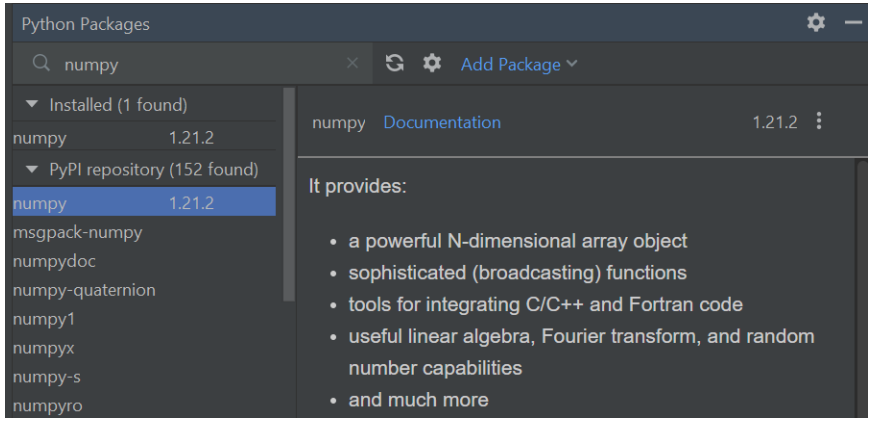
- Bottom bar of **Pycharm**



- Can use terminal to install packages to main IDE **Python** install



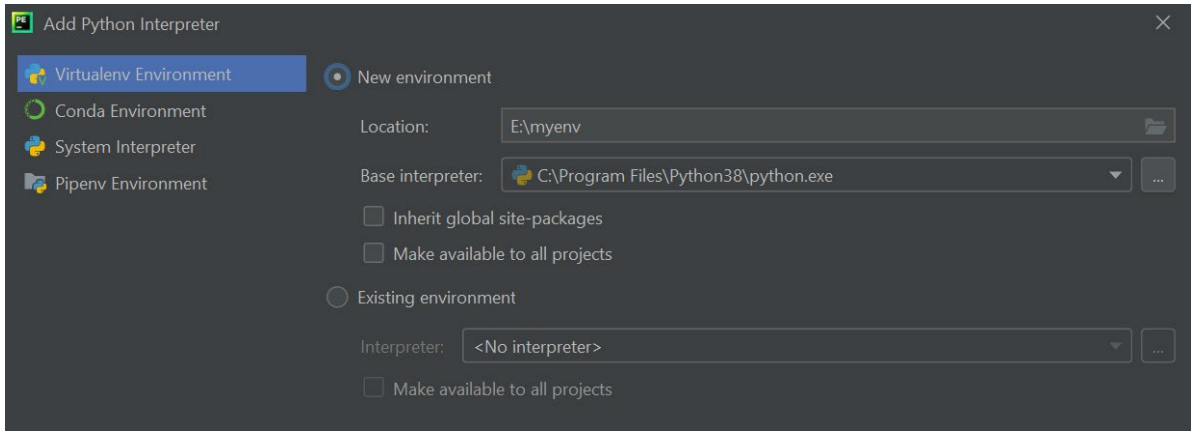
- Can also use package manager to do same



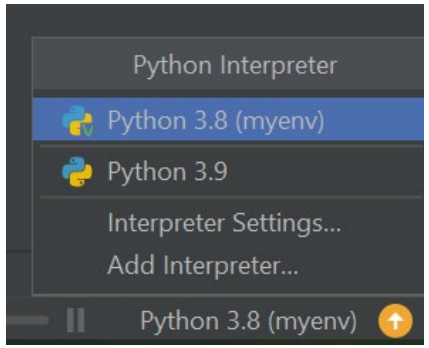
UNIVERSITY OF CALGARY

Data Science Installation (IDE-Pycharm)

- Instead of using system interpreter you can create virtual environment



- Bottom right corner you can toggled between **Python** interpreters



UNIVERSITY OF CALGARY

Data Science Installation (windows shell)

```
python -m venv myenv

.\myenv\Scripts\activate

pip install numpy
pip install pandas
pip install matplotlib
pip install seaborn
pip install ipython jupyterlab notebook

pip install scipy scikit-learn tensorflow keras statsmodels

deactivate
```

Data Science Installation (unix shell)

```
python -m venv myenv

source myenv/bin/activate

pip install numpy
pip install pandas
pip install matplotlib
pip install seaborn
pip install ipython jupyterlab notebook

pip install scipy scikit-learn tensorflow keras statsmodels

deactivate
```

Data Science Installation (docker)

- **Examples of data science oriented docker containers**
  - pytorch/pytorch — a simple container for Use Case 1 that includes Pytorch
  - jupyter/scipy-notebook — A container for Use Case 2 that includes Jupyter as the UI, and many python data science modules.
- **Many more advanced ones have been made**
  - DAGsHub/ml-workspace-minimal
    - Jupyter, JupyterLab, VSCode web-based IDE, Pytorch, Tensorflow, Sklearn, Pandas, and many other popular data science libraries & tools.
    - Full Linux desktop GUI accessible via a web browser, Easy terminal access via a web browser, Seamless Git integration optimized for notebooks
    - Integrated hardware & training monitoring via Tensorboard & Netdata.
    - Access from anywhere via Web, SSH, or VNC under a single port, Usable as a remote kernel (Jupyter) or remote machine (VSCode) via SSH

Custom DockerFile

Use a base image with the desired operating system and dependencies

FROM python:3.12

Set the working directory in the container (this version thinks it's a finished app)

WORKDIR /app

Copy the requirements.txt file and install dependencies

COPY requirements.txt .

RUN pip install -r requirements.txt

If you want to store project in container (often preference is instead to map volume from host instead as the COPY will be static code)

COPY .

If you want to set the command to run when the container starts (if you copied this file in with the prior step or have mapped volume access)

CMD ["python", "your_script.py"]

requirements.txt (example)

numpy==2.2.2
pandas==2.2.3
matplotlib==3.10.0
seaborn==0.13.2
ipython==8.32.0
jupyterlab==4.3.5
notebook==7.3.2
scipy==1.15.1
tensorflow==2.18.0
keras==3.8.0
statsmodels==0.14.4
scikit-learn==1.6.0

Importing Libraries in Python (style)

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import statsmodels as sm
```

Onward to … Libraries

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY