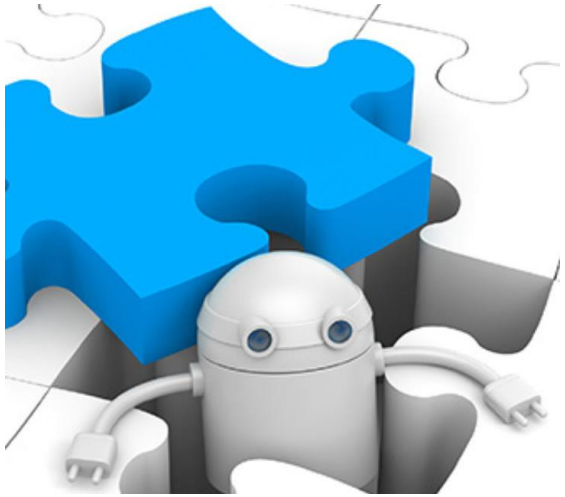
Tips and Pitfalls to Avoid when Teaching Machine Learning with Python to Chemical Engineers

Bryan R. Goldsmith

University of Michigan, Ann Arbor
Department of Chemical Engineering
2021 AIChE Annual Meeting
Session: Teaching Data Science to Students and Teachers II



Acknowledgements

Sharon C. Glotzer
Department Chair of Chemical Engineering at University of Michigan

Scott Fogler
(1939-2021)

MIDAS
MICHIGAN INSTITUTE FOR DATA SCIENCE
UNIVERSITY OF MICHIGAN

MICHIGAN INSTITUTE FOR COMPUTATIONAL DISCOVERY & ENGINEERING
UNIVERSITY OF MICHIGAN

CHEMICAL ENGINEERING
UNIVERSITY OF MICHIGAN

Materials
Modeling
Data Science
Descriptors
Catalysis
Goldsmith Lab

NSF

ACS

SCIENCE TO SHAPE THE FUTURE OF THE ARMY
ARMY RESEARCH OFFICE
1951

DEPARTMENT OF ENERGY
UNITED STATES OF AMERICA

Data science and machine learning are impacting many chemical engineering industries and research fields

- David Beck, James M. Carothers, Venkat R. Subramanian, and Jim Pfaendtner. "Data science: Accelerating innovation and discovery in chemical engineering." *AIChE J* (2016): 1402-1416.

Kitchin, John R. "Machine learning in catalysis." *Nature Catalysis* (2018).

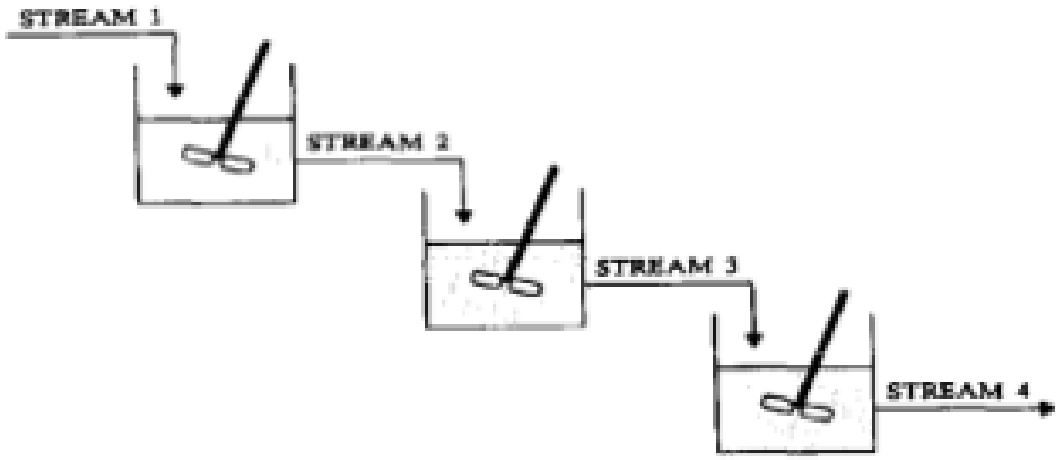
Chen, Hongming, *et al.* "The rise of deep learning in drug discovery." *Drug Discovery Today* (2018).

Data science in ChE has been around for a long time

Predict faults in chemical reactors

Table 1. List of selected faults

A	Inlet concentration of component A	Low
B	Inlet concentration of component A	High
C	Inlet flowrate	Low
D	Inlet flowrate	High
E	Temperature	Low
F	Temperature	High



Hoskins *et al.* "Artificial neural network models of knowledge representation in chemical engineering." (1988)

Reasons to believe data science and ML will have a bigger impact in ChE now than in previous eras.[1]

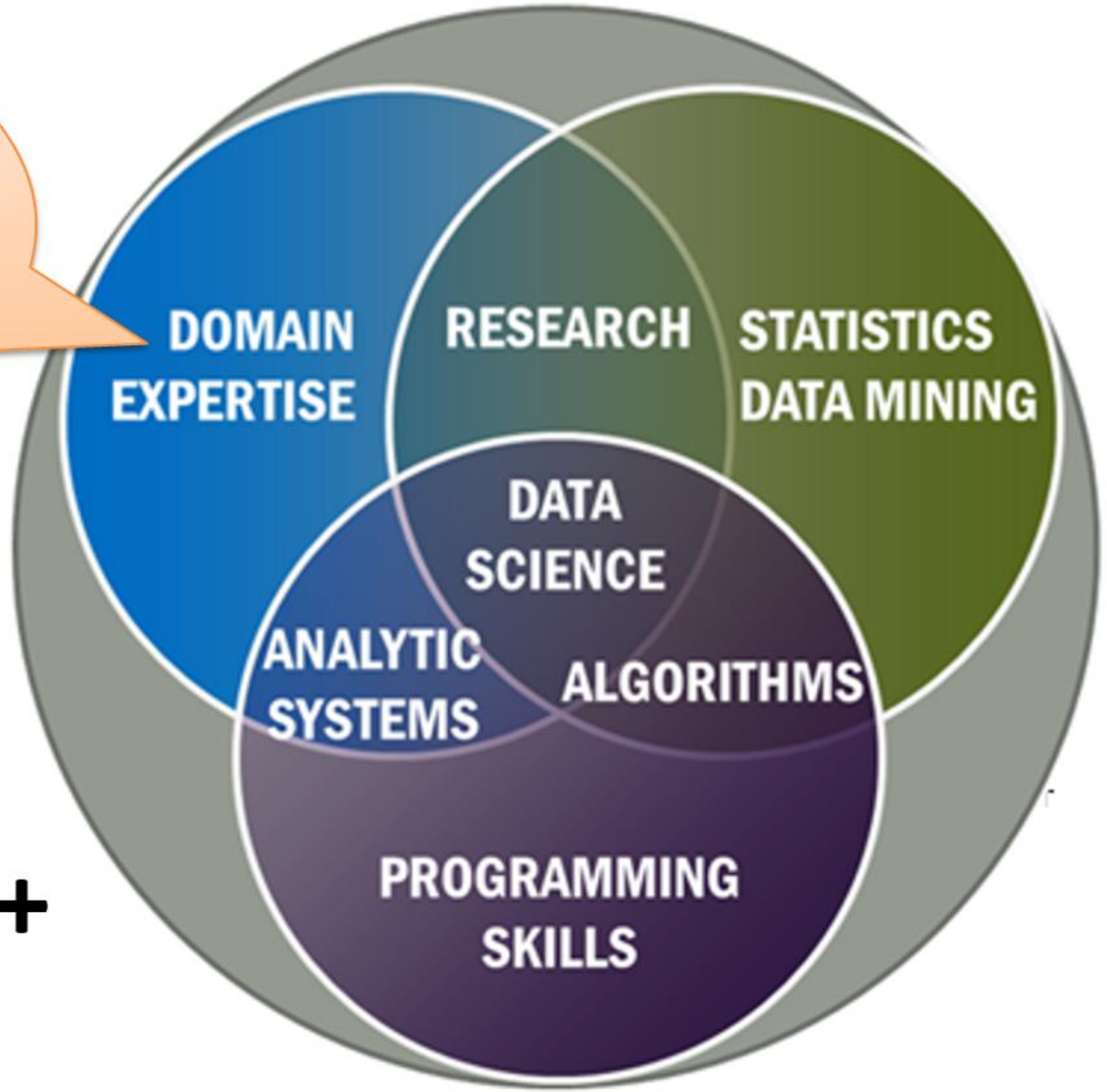
[1] Venkat Venkatasubramanian, "The promise of artificial intelligence in chemical engineering: Is it here, finally?" *AIChE J* (2019): 466-478.

Rise of big data, easy-to-use software, and availability of computing resources has been a boon to data science and ML

Chemical engineers

Domain Science

+



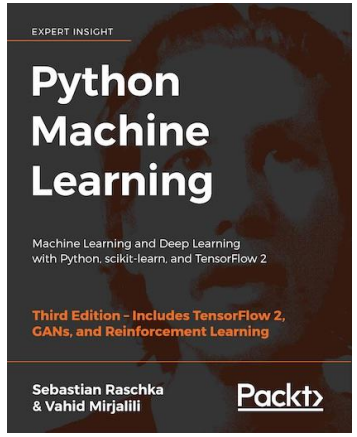
This talk will focus on sharing my experiences teaching graduate and undergraduate students data science and ML in ChE

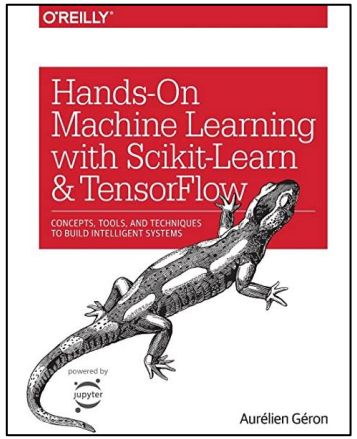
- **Created and taught thrice a graduate course on data science for chemical engineering students (open to students across engineering and senior undergrads).**
- Taught in F21, W20 and W19
- Course name: “Applied Data Science for Engineers”
- Averaging 28 students per class

- **Created and taught a first-year, freshmen level, course on data science and machine learning for engineering students.**
- W21, “Practical Data Science for Engineers” (47 students)

Graduate Level Course on Data Science for Engineers

SUPPLEMENTARY TEXT:





LECTURES: MW 9:00 AM – 10:30 PM

Taught in person, but lectures were recorded live through zoom and posted online.

COURSE OBJECTIVES

•Gain exposure to chemical engineering and related fields where data science and machine learning (ML) can play an important role.

•Learn tools and skillset to perform data science in your research and future jobs.

•Draw connections between theory, modeling, and applications in data science & ML.

•Learn ‘data acumen’.

•Provide opportunities for open-ended work.

•Practice and receive feedback on writing and communication.

Graduate Level Course on Data Science for Engineers

COURSE SUMMARY

- This course prepares engineering students to use data science tools during their master's and PhD thesis research as well as for post-graduation in industry, government, and academia.

- This course familiarizes students with the principles of modern data science techniques in the context of chemical engineering, materials science, and research.

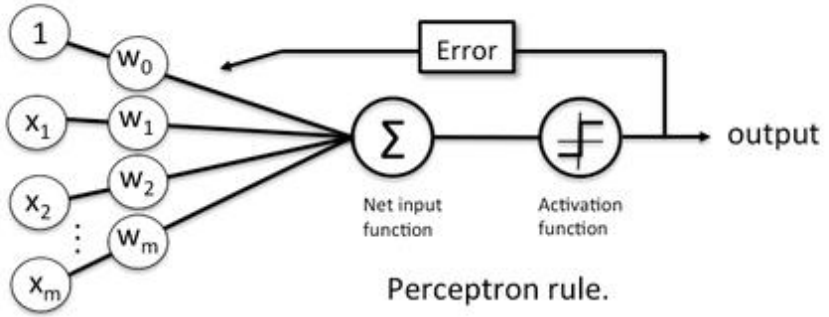
- Focus on an overview of data science tools used in engineering and science applications such as, *data curation, supervised and unsupervised machine learning, and data mining*. Algorithms covered include the *perceptron, PCA, kernel ridge regression, neural networks, random forests, support vector machines*.

- Homework exercises include hands-on practice of using data science.

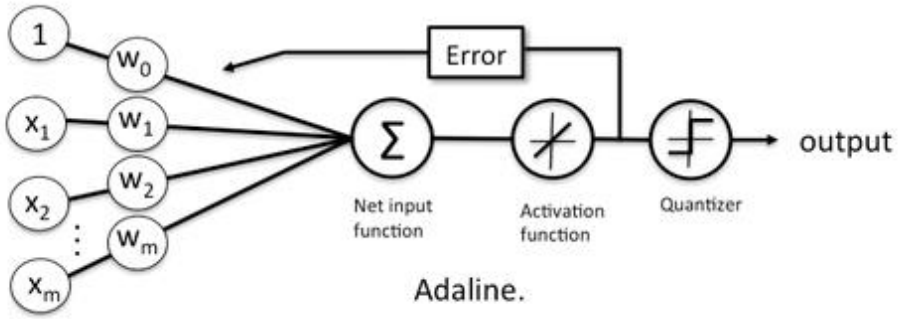
- Students are responsible for a data science project on a topic of interest.

Part of the Syllabus as an example

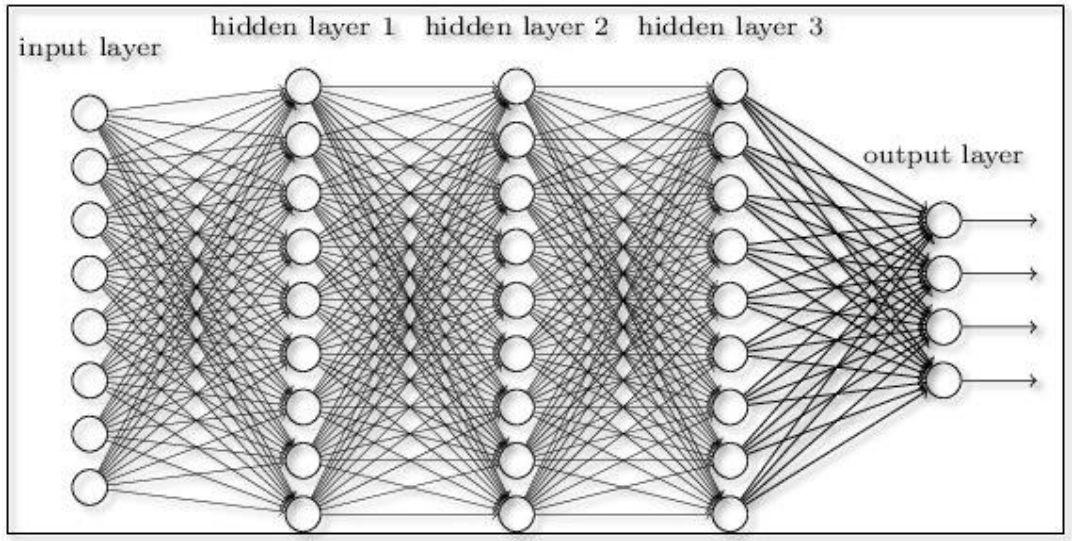
Date	Topic	Assignment Due
Monday 8/30	Introduction to the course and data science / ML	
Wednesday 9/1	Test set, training set, model validation, and features	
Monday 9/6	*Labor Day – NO CLASS*	
Wednesday 9/8	Simple machine learning algorithms for classification	
Monday 9/13	Tour of machine learning classifiers using scikit-learn	
Wednesday 9/15	Tour of machine learning classifiers using scikit-learn	**HW 1 (due Friday 9/17)**
Monday 9/20	Support vector machine classifier	
Wednesday 9/22	Logistic regression and KNN and Naïve Bayes	
Monday 9/27	Building good training sets – data curation and data leakage	
Wednesday 9/29	Building good training sets – data curation and data leakage	**HW 2 (due Friday 10/1)**
Monday 10/4	Dimensionality reduction techniques	
Wednesday 10/6	Predicting continuous target variables with regression	
Monday 10/11	Predicting continuous target variables with regression	
Wednesday 10/13	Decision trees and random forests	**HW 3 (due Friday 10/15)**
Monday 10/18	*Fall Break – NO CLASS*	
Wednesday 10/20	Schedule buffer	
Monday 10/25	Random forests and boosting algorithms	
Wednesday 10/27	Working with unlabeled data / some major software used in data science and 'big data'	**HW 4 (due Friday 10/29)**
Monday 11/1	Working with unlabeled data / some major software used in data science and 'big data'	
Wednesday 11/3	Neural networks, Tensorflow, and deep learning	**Send 1 pg. project update (Friday, 11/5)**
Monday 11/8 – Class will be taught remotely. Not in person. Gone at AIChE.	Neural networks, Tensorflow, and deep learning	
Wednesday 11/10 – Class will be taught	Neural networks, Tensorflow, and deep learning	**HW 5 (due**



Perceptron rule.

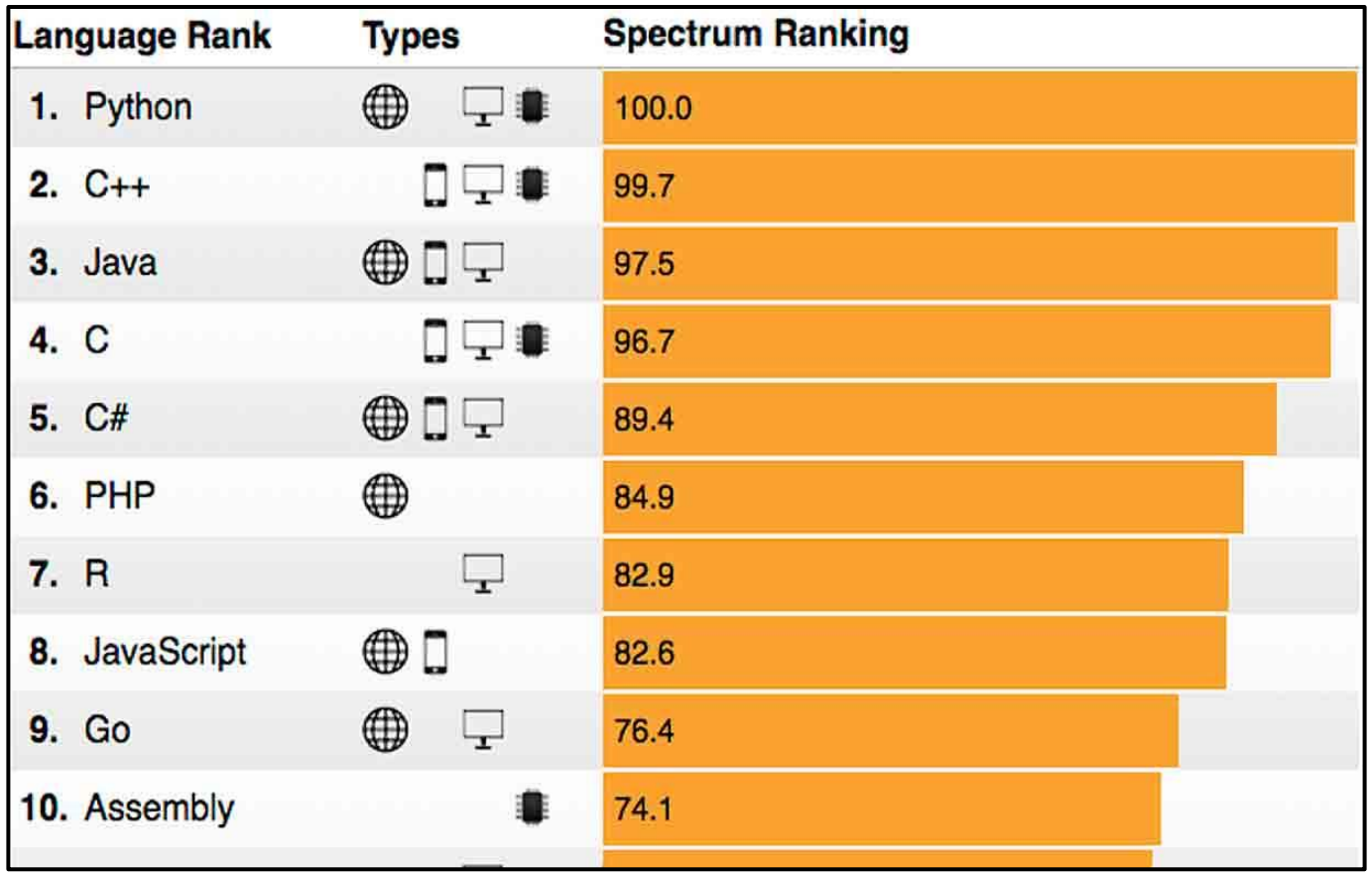


Adaline.



We use Python because it is one of the most popular languages and has a good data science eco-system

- From the IEEE Spectrum, 2018-07-31, ranking by typical IEEE member and Spectrum reader.



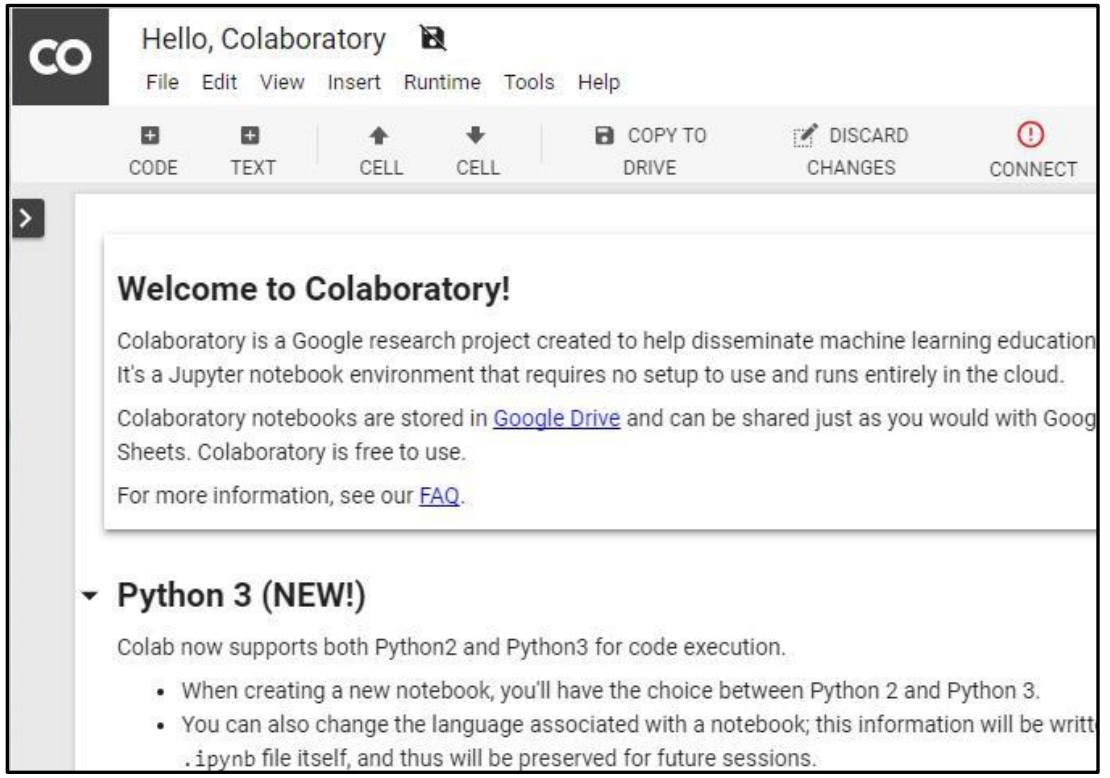
Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C++		99.7
3. Java		97.5
4. C		96.7
5. C#		89.4
6. PHP		84.9
7. R		82.9
8. JavaScript		82.6
9. Go		76.4
10. Assembly		74.1

We rely heavily on Google Colaboratory

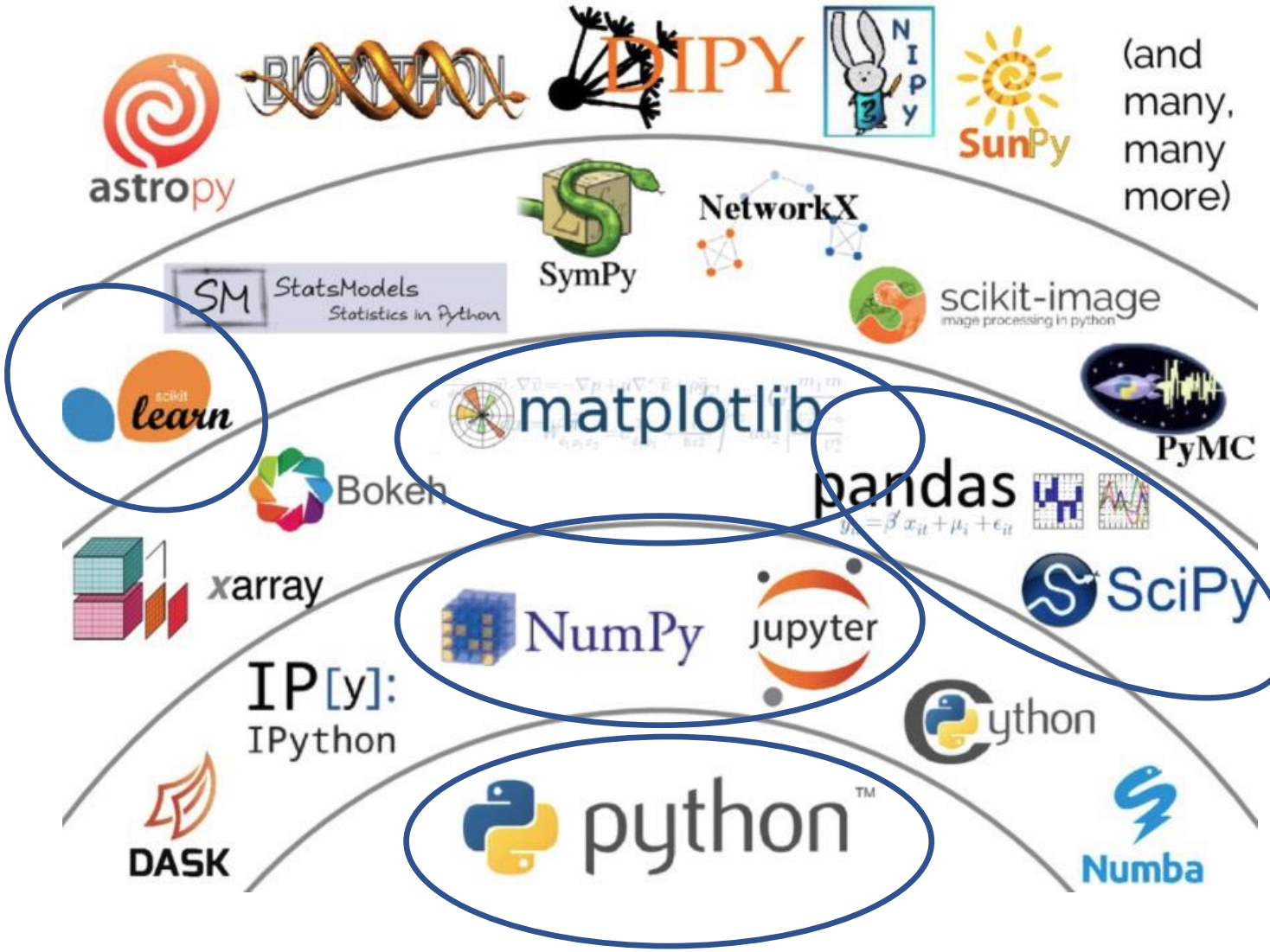
Colab Website: https://colab.research.google.com/

- Colaboratory allows you to write and execute Python in your browser, with
  - Zero configuration required
  - Free access to GPUs
  - Easy sharing

- Similar to how you can share and edit google word docs, google excel sheets, etc.



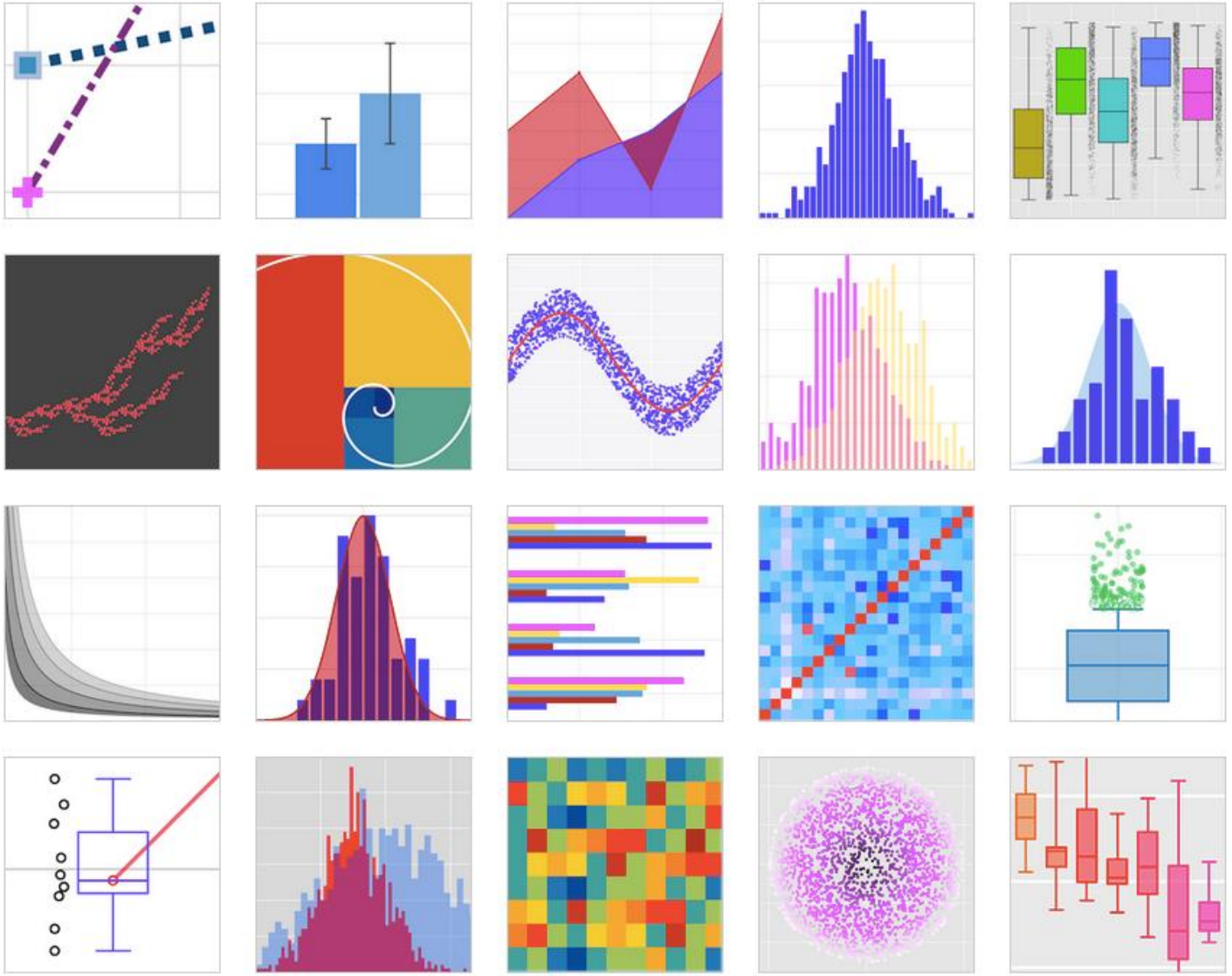
We cover many tools in Python's scientific stack



Python's Scientific Stack

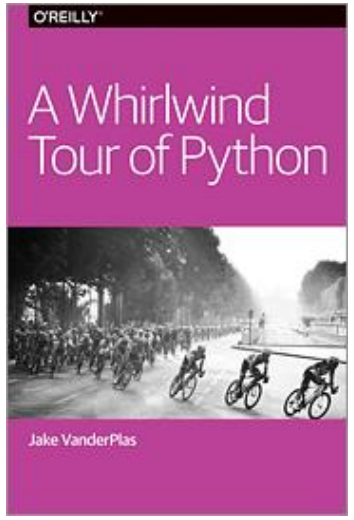
Image source: https://speakerdeck.com/jakevdp/the-unexpected-effectiveness-of-python-in-science?slide=52

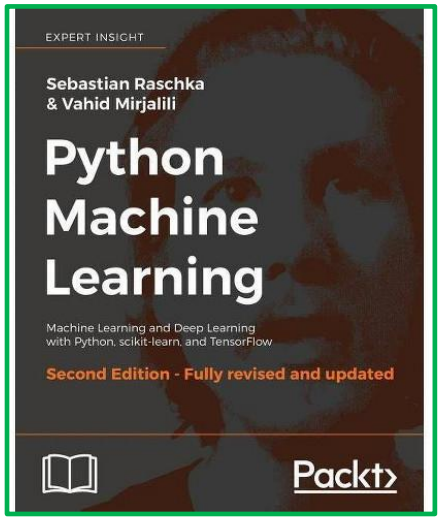
The students are taught Matplotlib because it is much more flexible than Excel and Origin plotting software and makes high-quality figures.



In the beginning of class, it is important to survey the students' programming and data science background. It is crucial to make available a lot of Python resources (example code, tutorials) to the students to fill any knowledge gaps.

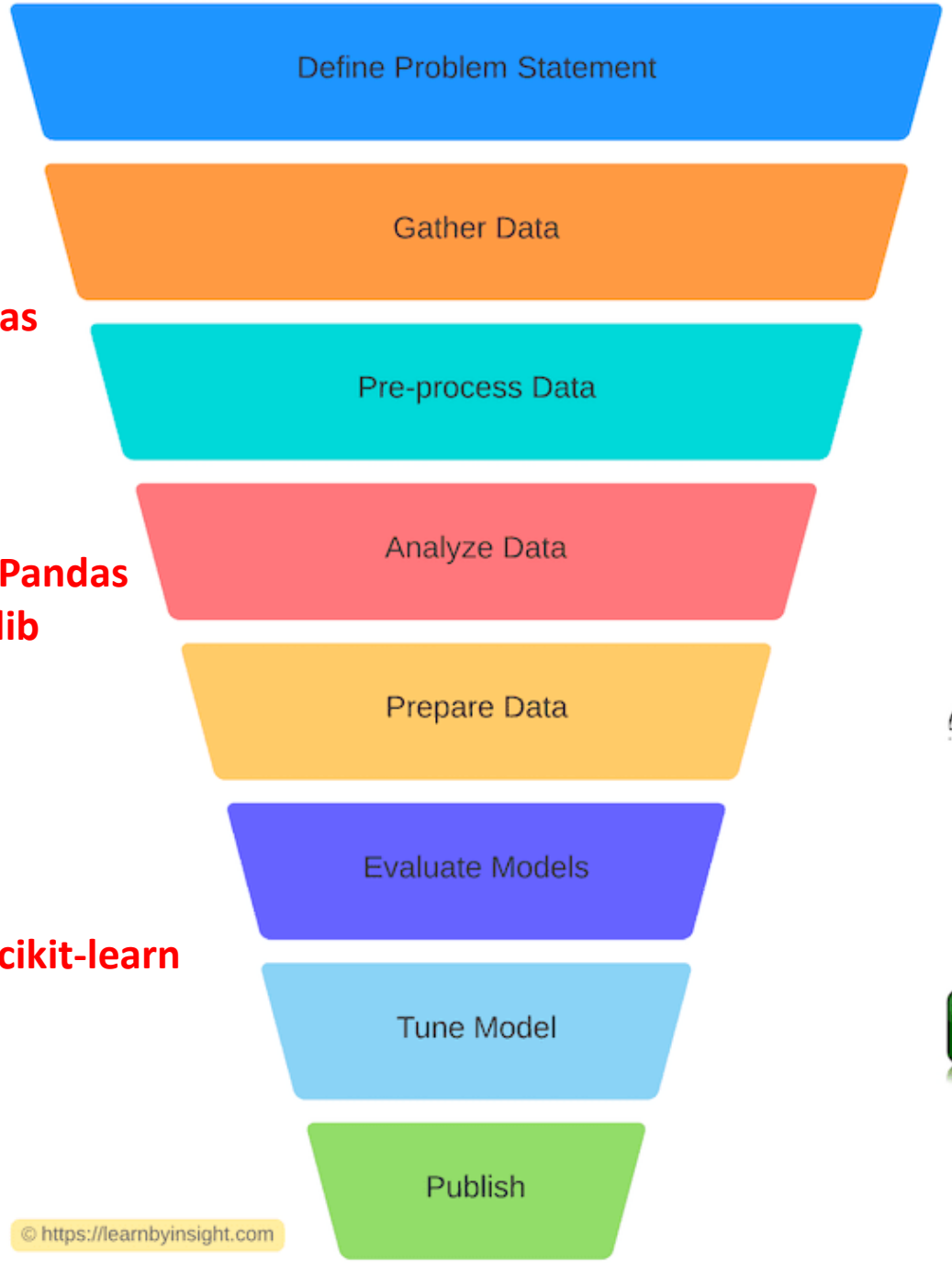
[1] Python Machine Learning (2nd edition) – Sebastian Raschka and Vahid Mirjalili
[2] A Student's guide to python for physical modeling – Jesse M. Kinder and Philip Nelson
[3] Hands-on machine Learning with Scikit-Learn and Tensorflow
[4] An introduction to statistical learning – Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
[5] Machine learning crash course: https://developers.google.com/machine-learning/crash-course/ml-intro
[6] Machine learning for absolute beginners, Oliver Theobald
[7] Pandas Introduction:
https://colab.research.google.com/notebooks/mlcc/intro_to_pandas.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=pandas-colab&hl=en
[8] Data Science (The MIT Press Essential Knowledge series), 2018, by John Kelleher and Brendan Tierney
[9] A Whirlwind Tour of Python https://jakevdp.github.io/WhirlwindTourOfPython/index.html





Has a lot of good example code to cover in class and to give for review

We systematically cover the full data science workflow



Pandas

NumPy, Pandas
Matplotlib

Scikit-learn

- *Problem Description*
- *Scope*
- *Feasibility*

- *Assumptions*
- *Constraints*
- *Provided data*

- *Data cleanup*
- *Data Imputation*
- *Drop*

- *Corelations*
- *Features of importance*
- *Type of problem*

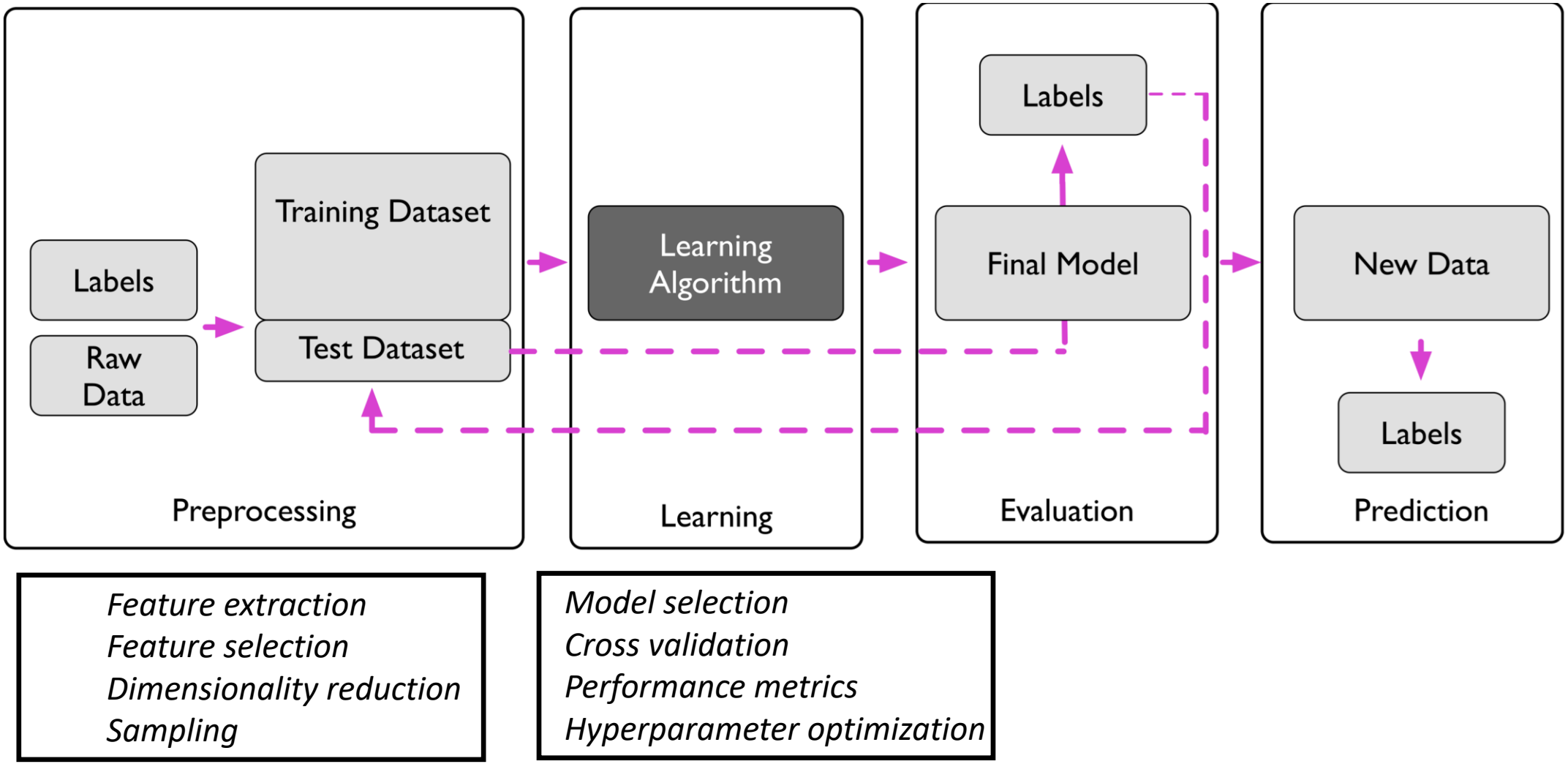
- *Transform*
- *Normalize*
- *Drop*

- *Train/Test*
- *Classify/Regress*
- *Feature reduction*

- *Cross validate*
- *Fine tune parameters*
- *Analyze results*

- *Solve*
- *Apply*
- *Learn more*

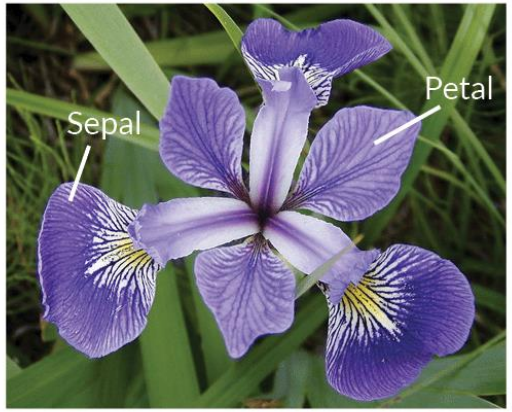
Students appreciate learning the “big picture” data science workflow



[1] Python Machine Learning (2nd edition) – Sebastian Raschka and Vahid Mirjalili

Concepts are introduced on simple traditional datasets but then shown in real research applications

Classic Iris Flower Dataset for Classification



Iris Versicolor

Iris Setosa

Iris Virginica

Food Safety by Using Machine Learning for Automatic Classification of Seeds

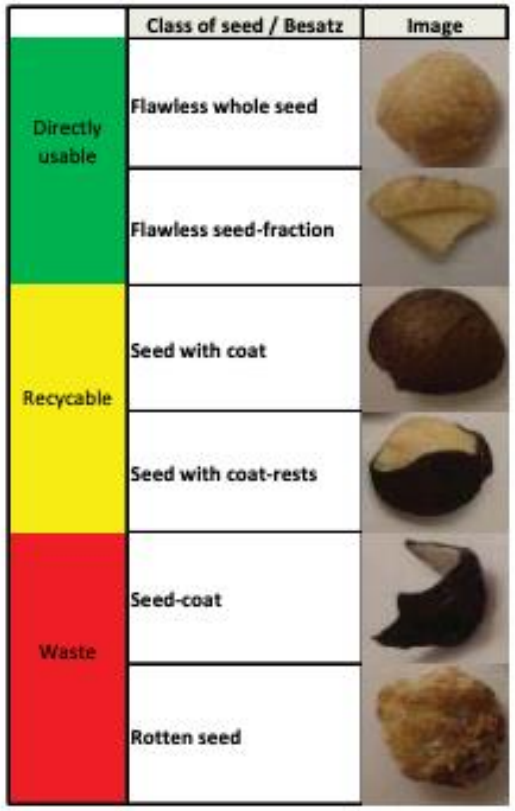


Figure 2. The 6 subclasses and the 3 superordinated classes of the Incanut

Lemanzyk, Thomas, et al. *Journal of Physics: Conference Series*. Vol. 588. No. 1. IOP Publishing, 2015.

Course Long Design Project

"To obtain hands-on practice doing data science, a major task in this course is to conduct a data-science project. This project is worth 40% of your overall grade. Pick a topic related to your research or an engineering/science subject of interest. Formulate a question(s) that can be addressed through data analysis and machine learning, then conduct your analysis and report your findings and conclusion in both a technical report and PowerPoint."

- I meet with all student teams in the beginning of semester to discuss projects and goals, as well as have a 1-page mid-semester project report to gauge progress.
- The students present their projects to their peers at the end of class and are judged by their peers. This is a fun event and all the students appreciated it.
- The students appreciate seeing previously successful projects from past years to clarify expectations.

Symbolic Regression and its Potential for Discovering Physical Laws

Olivia Liebman, Kyle Bushick, and Kevin Greenman

April 24, 2019

Predicting Biomass Yields from Global Crop Data

Jacob Saldinger, Isaiah Barth, and Cameran Beg

MATSCIE 593 / ChE 696 Applied Data Science for Engineers

GENERATIVE LEARNING OF THE TWO-DIMENSIONAL ISING MODEL NEAR CRITICALITY

CHE696 FINAL PROJECT

Jacques Esterhuizen
Department of Chemical Engineering
University of Michigan
Ann Arbor, MI
esterhui@umich.edu

Frank Doherty
Department of Chemical Engineering
University of Michigan
Ann Arbor, MI
fdoherty@umich.edu

Nathan Ng
Department of Chemical Engineering
University of Michigan
Ann Arbor, MI
nsng@umich.edu

April 24, 2019

Some takeaways on the graduate data science course

1) My lectures were split into 2/5 traditional lecturing (notes with gaps), 2/5 covering code via Google Colab, and 1/5 hands-on practice of concepts.
2) Homework was both conceptual and had *hands-on programming* exercises.
3) Students enjoyed working on semester-long data science projects of their choice.
4) Chemical engineering students were not necessarily interested in nitty gritty computer science details, but rather by the applications of data science and ML tools to industry and research problems.
5) There was significant interest from students in bioengineering, pharmaceutical sciences and materials science departments, not only chemical engineers.
6) We relied heavily on *Google Colab*.
7) This class served as a survey course. For most graduate students, this was their first experience with python, data science, and ML.
8) Very positive feedback all three years on the course.

Teaching *freshmen*, undergraduates, data science

- More challenging than teaching graduate students because the students do not have linear algebra, statistics, or much programming background.
- I used the graduate class as a template for this course and removed as much math details as possible. Assumed intro calculus knowledge.
- Important, this class had a weekly lab component (3 hrs) on top of 3 hours of lecture, so Python topics can be covered in detail in labs.
- The labs were team based and had hand-outs of goals for each lab.
- The class had a semester-long data science project that was *well-defined*.

Example lab 1 handout

Lab Assignment 1

Introduction to Python Programming, Google Colab, and Jupyter Notebook

Background

In this hands-on "laboratory", we will learn the essentials about Python and other open-software tools, which will enable us to perform data science and machine learning of complex datasets. This lab will focus on:

1. Covering essentials of Python Programming.
2. Introducing the Google Colab environment for programming in Python.
3. Introducing Anaconda, an open-source distribution of the Python, and Jupyter Notebooks.

Go to the following Google Colab link to follow the tutorial with Prof. Goldsmith and team (or download it from Canvas). Make a copy of the Google Colab Notebook and upload to your Google Colab, so then you can edit your own version and follow along.

Google Colab Link: https://colab.research.google.com/drive/1MBri84asV935ZyGHum2M-k3N_MMIKhFR?usp=sharing

At the end of the hands-on lab tutorial, answer the following questions in a Google Colab or Jupyter Lab Notebook. The solution to some of these questions will eventually be requested in your Lab Report 1 (to be posted later).

Questions to answer after completing the hands-on lab tutorial

1. Why is Python a powerful and useful programming language?

2. Which code is usually faster to execute, code written in Python or C++?

3. In Python, what is `777 // 3`? What is `73 ** 3`? What is `15393 % 2`?

4. What is the final value of x for the following code?

```
x = 7
x += 8
x **= 5
```

5. Write code in Python to compute the square numbers of [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] and print only the odd numbers.

6. In Python, define variables that you need to calculate a value from your favorite equation, and then make the calculation referencing those variables. Use comments to identify the equation and to note the units of all variables you defined.

7. Write and execute a code that reports the square root of `array([0, 1, 2, 3, 4, 5, 6, 7, 8])`. Show the code input and output.

8. Create a 1D array of numbers from 0 to 9. Convert this 1D array to a 2 × 5 array. Show the code input and output.

9. In Python, create a list of at least five things or people (represented in string form) that you are grateful for. Sort it and print it. Then add five more things, reverse sort, and print again. Also print the length to make sure you have ten. Show the code input and output.

Example Lab Handout toward the end of course on Neural Networks

- I would often give partially completed "skeleton" code to help guide the students in labs. In homework, they would leverage the labs to advance their knowledge further.

Practice with Feedforward Neural Networks

Background

In this lab, we will gain hands-on practice implementing Feedforward Neural Networks. We will cover:

- Implementing a feedforward neural network (aka multilayer perceptron) from scratch
- Training and testing a feedforward neural network using scikit-learn

See Lab_10.ipynb on Canvas or use the shared Google Colab link and download it to follow the tutorial with Prof. Goldsmith and team:
https://colab.research.google.com/drive/1pgqik4eIOZIEMJckT0kC3oxnAYCFPyTX?usp=sharing

At the end of this lab, answer the following questions in a Google Colab or Jupyter Lab Notebook. The solutions to some of these questions are requested in your Lab Report 3.

Questions to Answer After Completing Lab_10.ipynb

1. Train your own multilayer perceptron (MLP) using sklearn on the MNIST dataset. See 'from sklearn.neural_network import MLPClassifier' and accompanying documentation. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

a. Create the training and test set for MNIST data (first 60,000 samples are training, and last 10,000 samples are for testing).

b. Fit your MLPClassifer using 'hidden_layer_sizes' of (10,1), (100, 1) and (50, 2). Use the 'relu' activation function, the 'adam' solver, set 'max_iter' to be 100, and use 'validation_fraction = 0.3'. Set 'verbose = true' if want to see the loss at each epoch. What does (50, 2) mean in terms of neural network architecture? Note: it will take a few minutes to train the models, because the dataset is moderately large.

c. Plot the loss curve (see '.loss_curve_' attribute) for *each* of the three models built using different 'hidden_layer_sizes'. Properly label the x and y-axes.

d. Report the 'accuracy_score' for each of the three MLPClassifer models on both the training set and the test set. Which of these three models performed the best in terms of accuracy? How does the best model compare to the MLP model that we implemented from scratch?

Creating Data Science Projects for undergraduates

kaggle

- I took online datasets from Kaggle and anonymized it. For the first year, I did wind and solar energy.

Engineering 100 Design Project Prompt

An international solar panel company is collaborating with a weather agency in Honolulu, HI to analyze data on solar irradiation. The solar company would like to build a solar forecasting model to predict Solar Radiation vs. Date (Time). Your engineering consulting company has been tasked to analyze a dataset from the weather agency and answer a set of the client's questions, as well as formulate your own analysis.

Goal: Build a predictive model for Solar Radiation vs. Date/Time to improve sunlight prediction capabilities, and to understand what key factors affect solar radiation.

Context of Solar Panels: Solar cells are made up of two semiconductors that are designed to generate an electric current when light is absorbed. Most modern commercial solar panels use two slabs of doped silicon, one *p*-type and the other *n*-type. The *p*-type silicon is doped with electron-deficient elements (e.g., boron) which create electron holes, whereas *n*-type silicon is doped with electron-rich elements (e.g. phosphorous) which create free electrons within the silicon crystal.[1]

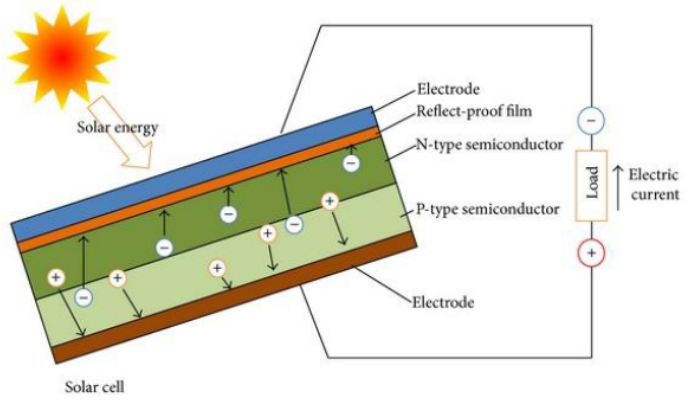


Figure 1. Diagram showing how electricity is generated in a solar panel.[2]

Engineering 100 Design Project Prompt Notes

A Wind Turbine Company located in Spearville, KS, USA is looking to build its next wind turbine farm. They have collected data from one of their wind turbine farms and wish to build a predictive model for wind turbine power generation, as well as understand what key factors affect wind turbine power generation. Your engineering consulting company has been tasked to analyze a dataset from the client and answer a set of the client's questions, as well as formulate your own analysis.

Goal

Analyze dataset and predict wind turbine power production from the wind speed, wind direction, month of the year and the hour of the day.

Overview of Wind Turbine Systems

Wind turbines generate power by converting the mechanical energy of moving air into electrical energy. The blades of the turbine allow wind to create a torque that drives an electrical generator.

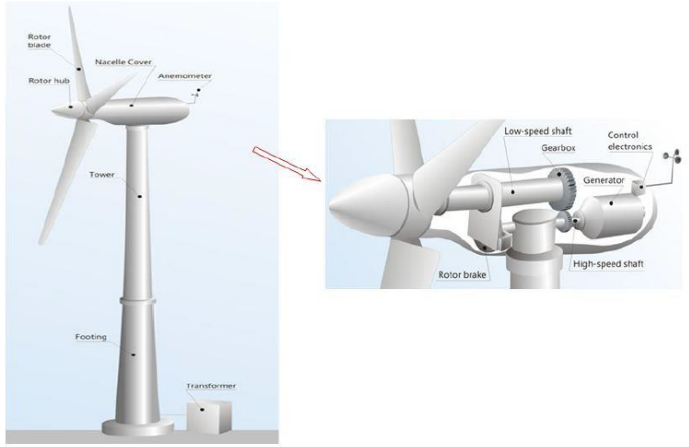


Figure 1: Basic schematic of a wind turbine.[1]

Final Thoughts

- **ChEs are interested in taking data science and ML classes** that show they how to use tools that will benefit their research or future work in industry.
- By focusing on domain applications in ChE and **cutting out non-critical CS material**, we can create an engaging and relevant course for ChEs.
- **Course long, team-based, data science projects** are a great way to reinforce lecture concepts and have the students use what they learned in lecture and labs
- **Labs and hands-on coding exercises are needed** to reinforce concepts.
- **Can skip some of the math** - Undergrads can learn and appreciate using supervised and unsupervised ML models hands-on without understanding all the math.
- **Give a lot of resources early on for Python programming.** Most students lack strong programming background.
- **Need better ChE datasets**. More collaboration with Industry would be helpful.

Questions?

If you would like to see some of my course materials, email me at bgoldsm@umich.edu