GENERAL ASSEMBLY

Intuit Welcome to Data Science

November 2, 2020

Intuit Classroom Norms

- Your first impulse with a question should be to post it in the Slack #classroom channel. Speak up if it's urgent.
    - If it gets out of hand we'll ask everyone to stay on mute until invited to come off mute. In the beginning though let's see if we can't be orderly without too many rules and unnecessary interruptions.
- Teng and Meggan will answer questions posted in Slack as they arise. And bring to my attention any questions that may help the entire group.
- DM the whole instructional team. We'll address it appropriately. This may include going to a breakout room.
- If you feel that you need an opportunity to speak with a member of the instructional team further, please schedule office hours.

Learning Objectives

- **Understand** what data science is and what it isn't.
- **Understand** broadly what data scientists do.
- **Explain** the data science workflow.
- **Create and update** a git repository and sync it with GitHub.
- **Understand and explain** data types in Python
- **Understand and explain** namespaces in Python
- **Understand** basic operations in Python

What is Data Science

What is Data Science

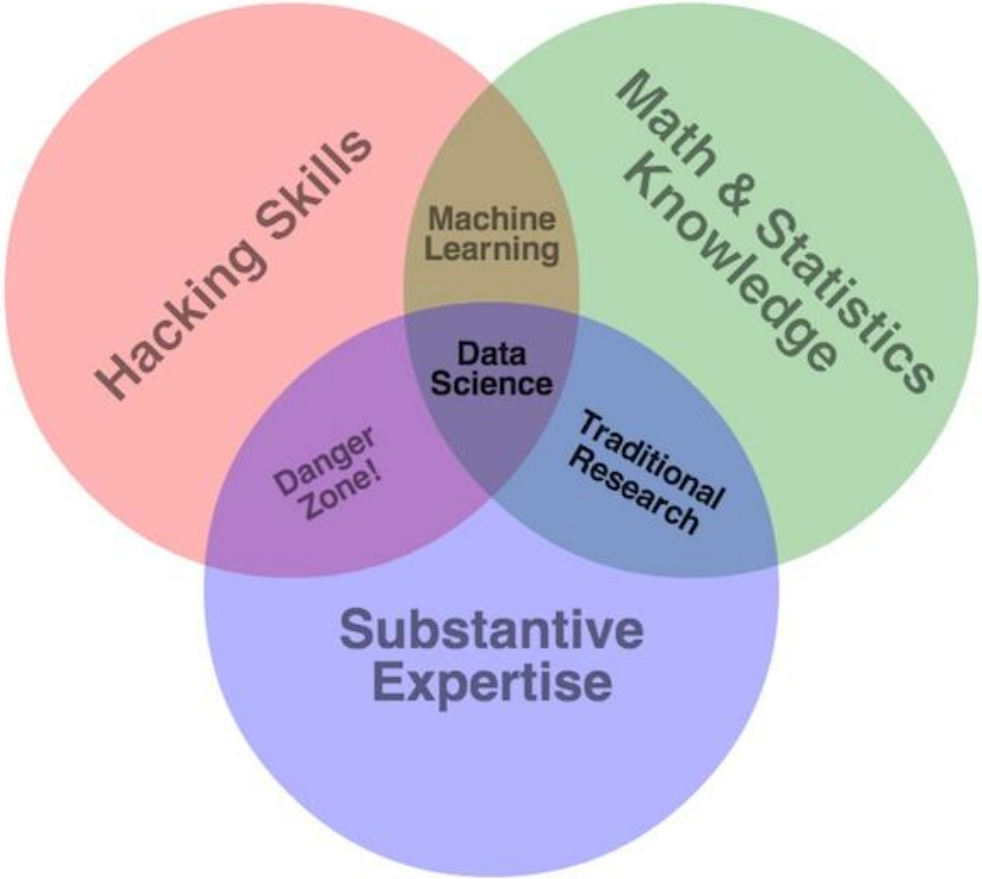
“Data Science refers to an area of work concerned with the collection, preparation, analysis, visualization, management and preservation of large collections of information.”

-- Jeffrey Stanton
An Introduction to Data Science
Syracuse University School of Information Studies

What is Data Science

Name 5 products or services that use or are built on data science?

Who is a Data Scientist?



What is a Data Scientist

“A data scientist is someone who can obtain, scrub, explore, model and interpret data, blending hacking, statistics and machine learning. Data scientists not only are adept at working with data, but appreciate data itself as a first-class product.”

-- Hilary Mason, chief scientist at bit.ly

A Simple Data Science Pipeline

1. **Prepare to Run A Model**
   a. gather, clean, integrate, restructure, transform, load, filter, delete, combine, merge, verify, extract, & shape.
2. **Run the Model**
3. **Communicate the Results**

Data Science Workflow (one approach)

Doing Data Science: The GA Process

1. **Frame:** Develop a hypothesis-driven approach to your analysis. Ask a good question.
2. **Prepare:** Select, import, explore, and clean your data.
3. **Analyze:** Structure, visualize, and complete your analysis.
4. **Interpret:** Derive recommendations and business decisions.
5. **Communicate:** Present insights from your analysis to different audiences.

Data Science: Asking Questions

SMART Goals Framework

- **Specific:** The data set & key variables are clearly defined.
- **Measurable** The type of analysis & major assumptions are articulated.
- **Attainable** The question you are asking is feasible and unbiased for the data.
- **Reproducible** The project can be understood and reproduced.
- **Time-bound** The time period for the problem

What are some questions asked in data science (general and specific)?

Data Science is about Data Products

*A **data product** is a product that facilitates an end goal through the use of data.*

*Data science is about **building data products**, not just answering questions.*

*Data products **empower** others to use the data.*

Online Databases
- Raw Data
- Derived Data

Algorithms as a Service
- Google Image
- GPT-3

Dashboards & Visualizations
- Google Analytics

Data-driven apps
- Autonomous Vehicles
- Netflix Recommendations

Data Science Exercise

Task: You work for a real estate company and they want you to develop a data science application to determine the best properties to buy and sell. Look at the Ames data dictionary and come up with some questions.

Frame - what question would you ask?

Prepare - what data would you need?

Analyze - how would you build a model and what would we predict?

Interpret - How would you draw conclusions or determine next steps?

Communicate - How would you communicate results to your boss?

MACHINE LEARNING TERMINOLOGY

Supervised learning (a.k.a., "predictive modeling"):

- *Classification and regression*
- Predicts an outcome based on input data
  - Example: Predicts whether an email is spam or ham.
- Attempts to generalize
- Requires past data on the element we want to predict (the target)

Unsupervised learning:

- *Clustering and dimensionality reduction*
- Extracts structure from data
  - Example: Segmenting grocery store shoppers into "clusters" that exhibit similar behaviors.
- Attempts to represent
- Does not require past data on the element we want to predict.

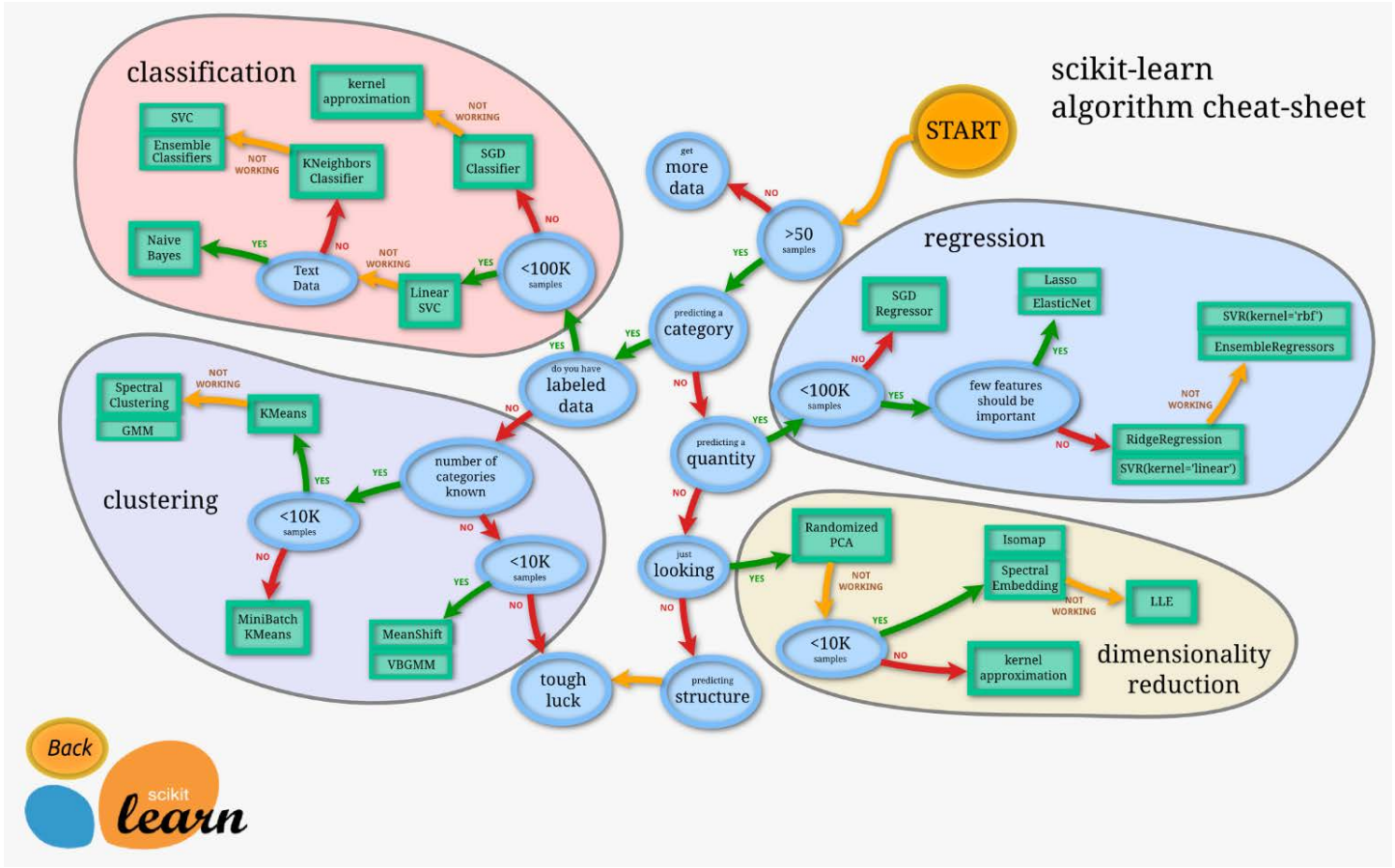
USING MACHINE LEARNING

- Oftentimes, we may combine both types of machine learning in a project to reduce the cost of data collection by learning a better representation. This is referred to as transfer learning learning.

- Unsupervised learning tends to present more difficult problems because its goals are amorphous.

- Supervised learning has goals that are almost too clear and can lead people into the trap of optimizing metrics without considering business value.

Slack Exercise – Let's explore a few areas of Data Science we'll be learning in class!



SUPERVISED LEARNING

Most frequent type of work that data scientists do and will be the main focus of this course. How does supervised learning work?

1. We train a **machine learning model** (more on that shortly) using **labeled data** (the "response" label from earlier).

2. We make predictions on **new data** for which the response is unknown.

The primary goal of supervised learning is to build a model that “generalizes” —i.e., accurately predicts the **future** rather than the **past**!

CLASSIFICATION VS. REGRESSION

There are two categories of supervised learning:

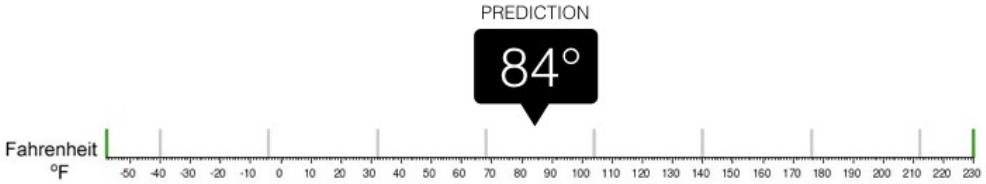
- **Regression**
  - The outcome we are trying to predict is a continuous value.
  - Can you think of anything we would want to predict like this?

- **Classification**
  - The outcome we are trying to predict is categorical (i.e., it comes in one of a set number of classes).
  - Can you think of anything that we would want to predict like this?

The type of supervised learning problem has nothing to do with the features; only the response matters!

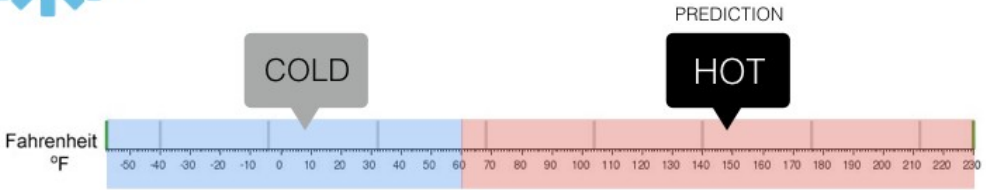
Regression

What is the temperature going to be tomorrow?



Classification

Will it be Cold or Hot tomorrow?



UNSUPERVISED LEARNING

Common Types of Unsupervised Learning

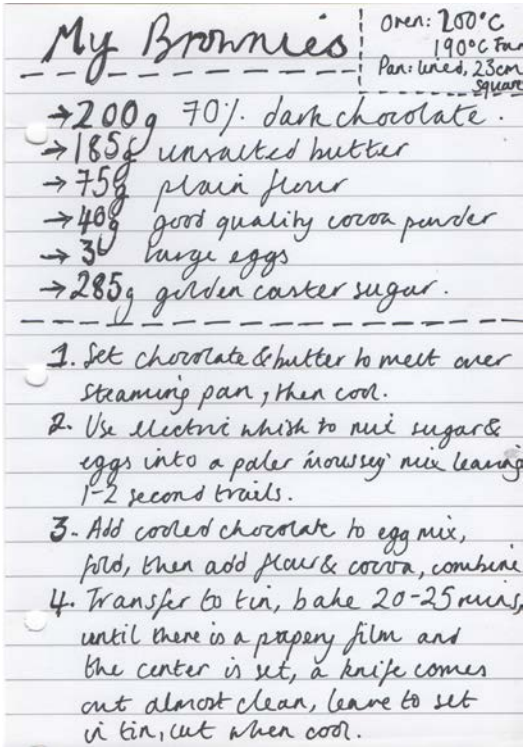
- **Clustering:** Groups "similar" data points together.
- **Dimensionality reduction** : Reduce the dimensionality of a data set by extracting features that capture most of the variance in the data.

Types of Customers at a Bar

- **Observations:** Customers.
- **Features**: Drink purchases, people they interact with, etc.
- **Response or target variable:** There isn't one —instead, we group similar customers together.

Understanding algorithms

As defined: a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.



Think of it like a recipe. A Step-by-step process needed to get a successful outcome.

- There are many ready to use
- You may need to modify it
- Often you'll want to build your own from scratch

Git & GitHub

Git vs GitHub

Git (2005): Version control system.

- Responsive, easy to use, and free
- Branching - create independent branches to try out ideas.

GitHub: Cloud-based service to host git repositories.

- Allows coordination for small and large teams.
- Branching allows team members to work on updates without overwriting.
- Merge branches safely.

Git Commands

Git is both complex and simple. Focus on the primary commands:

`git init` - to initialize a git repo.

`git add` - add the file to the local repo.

`git commit -m “wrote commit message”` - create snapshot of local repo.

`git push` - push the local repo to GitHub

`git pull` - pull the changes in GitHub repo to the local repo

`git branch` - create or switch to a branch

`git status` - check the current status of a local repo.

`git remote add` - add remote connection for local repo

Git Guided Exercise 1 - Fork a repo

1. Define a working directory for this class. All your work will live under this directory. Example: `mkdir dat-intuit`
2. Under this directory create a folder named git-demo
3. In your browser go to the class repo git-demo (link in the Slack)
4. Click the “fork” button
5. On your GitHub repo copy the URL (“clone or download”)
6. On the command line: `git clone <URL.git>`
7. Add a new file (e.g. touch new_file.txt)
8. `git add new_file.txt`
9. `git commit -m “add new_file.txt”`
10. `git push`

Git Guided Exercise 2 - create a new repo

1. Create a new directory submission and cd into this directory
2. Create a new file in this directory (e.g. touch submission.txt)
3. Type git init
4. git status
5. git add submission.txt
6. git commit -m “add submission.txt”
7. git status
8. On GitHub create a corresponding repo, submission.
9. git remote add origin <URL.git>
10. git push -u origin master

Git Guided Exercise 3 - Branching & Pull Requests

1. In the directory hello-intuit type git branch <branch name>
2. git checkout <branch name>
3. add newer_file.txt (e.g. touch newer_file.txt)
4. git add .
5. git commit -m “add newer_file”
6. git push --set-upstream origin <branch name>
7. Go to repo on GitHub
8. Select “Compare & pull request” or New Pull Request
9. Click create pull request
10. Click merge pull request
11. Click confirm merge

Python Foundations

Learning Objectives: Python

This portion of the class is all about Python. By the end you will be able to:

- Explain Python importing & namespaces
- Explain various Python data types, formatting, code blocks and indentation
- Explain and use control flow operations in Python
- Apply the Python to practical problems

Examples and activities will be in the context of data science.

Intuit Data Science

Python Basics Review

Python Data Types - Review

- **Immutable**
    - Numeric
    - Strings
    - Tuples
- **Mutable**
    - Lists
    - Sets
    - Dictionaries

Python Data Types - Review

- numeric - float (6.23, 0.939), int (0, 12, 939), complex (2 + 4j), bool (True, False)
- Strings - “This is a string.” So is ‘ajAua#47A9!@1920’
- Lists - [1, 2, 3], [‘car’, ‘boat’, ‘plane’], [1, ‘car’, 3, [2, ‘boat]]
- Tuples - (3, 5, 7, 3)
- Sets - set(1, 1, 2, 3, 4, 3, 4) -> (1,2,3,4) - elements must be immutable
- Dictionaries - {key: value} - {‘1’: ‘car’, ‘2’: ‘boat’, ‘3’: ‘plane’, ‘bike’: 4}
    - Key is always a string, value can be any legitimate Python value.

Intuit Data Science

Python Structure

Formatting & Modules

Python Modules

- Modules are logical groupings of code
    - Often comprised of many, typically related methods and functions
    - Simple as a python file *intuit-code.py.*
    - Complicated as a whole library *sklearn or pandas.*
- Why modules
    - Don’t need all capabilities all the time
    - Write new code and add easily
- Accessed via **`import`** statement
    - Namespace - avoids conflicts

Python Formatting - Whitespace, Indentation, & Code Blocks

Whitespace is use to delimit blocks of code and separate individual items on a line.
These two statements are identical

```
a = [   [1,2,3], [4,5,6], [7,8,9]   ]
```

```
a = [   [1,2,3],
        [4,5,6],
        [7,8,9]   ]
```

Whitespace & indentation defines what code belongs to which block - defines execution flow. PEP8 recommends 4 spaces for indentation, not tabs.

```
total = 0
for i in range(1,6):
    print(" in the loop", i)
    total += i    # add the next value

print("\n out of the loop", total)
```

Intuit Data Science

Python: Control Flow

Python Control Flow: Learning Objectives

- Understand & explain control flow and conditional programming
- Explain & implement primary control operators and how each works?
  - Implement for and while loops
  - Use if...else statements
- Combine control flow and conditional statements
- Create functions to perform repetitive actions.
- Understand and explain *Truthiness* in Python.

Python: Primary Control Flow Operators

Loops:

- **if...else** - If this, then that, else something else.
- **for -** repeat an operation for every item in an object
- **while -** repeat an operation as long as a condition is met

- **continue-** go to the next iteration in the loop immediately
- **break** - quit the loop immediately

Guided Walk-Through:
Python Control Flow Walk-through

A simple example:

```
colors = [“red”, “green”, “blue”]
# for loop
for i in range(4):
    print(f“color = {colors[i]}”)

# easier and faster
for color in colors:
    print(f“color = {colors}”)
```

Intuit Data Science

Python: Functions

Python Functions: Learning Objectives

- What is Python function?
- What does it enable us to do?
- How to write a Python function?
- What are some tips and best practices for writing Python functions?

Python Function: Definition

- **Definition**: a rule for taking 0 or more inputs (called arguments) and returning a corresponding output. A block of organized, reusable code used to perform a single task or action.
- **Example:**

```
def double(x):
    """Double the value provided in the
     argument. This is a docstring. It is used to
     explain what the function does.
     Args:
        x : value to be doubled
    Returns: output - the value in x doubled
    """

    output = x * 2
    return output
```


GA