Intro to Data Manipulation & machine Learning

Vince Choi & Bowen Liu

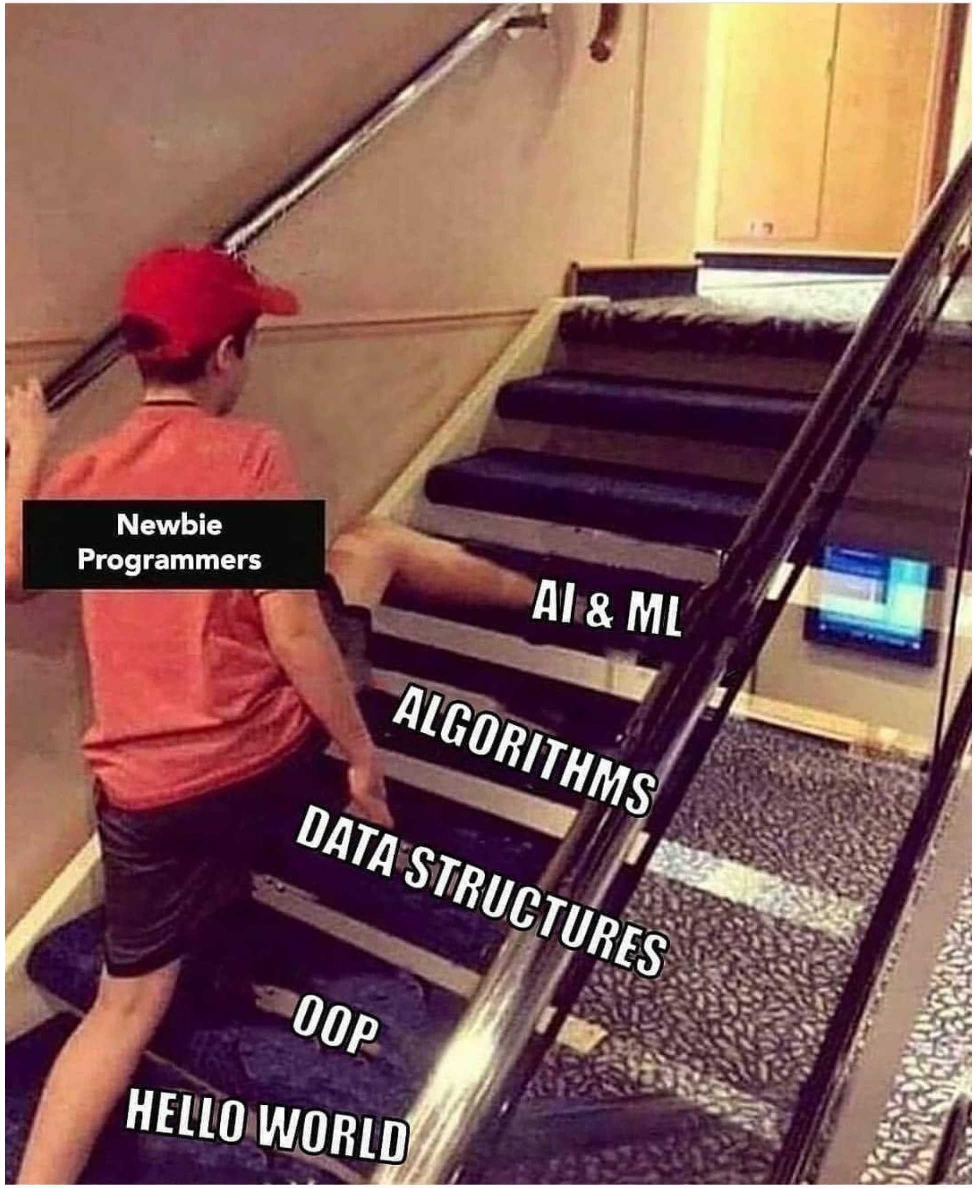
DSI

Data Science Innovated

Learn Share Grow

From Novice to Pro

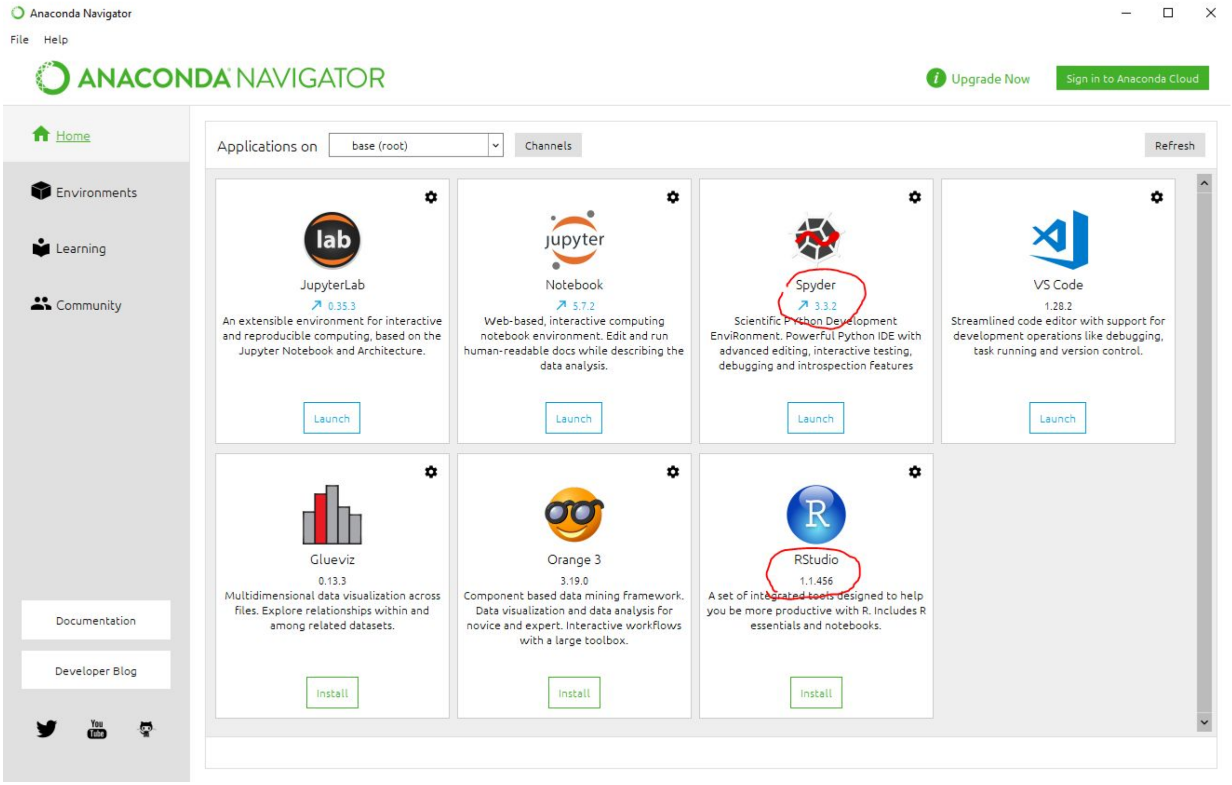
- Environment and Coding
  - Recap on Python and R
    - General syntax and terminology
    - Packages
    - Crash course
  - What to do with data
    - Manipulation
    - Visualization
- Intro to Machine Learning (ML)
  - Plotting
  - Clustering (unsupervised)



Environment

Resources for this workshop: https://github.com/dsinnovated

Setting up



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

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

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

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

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

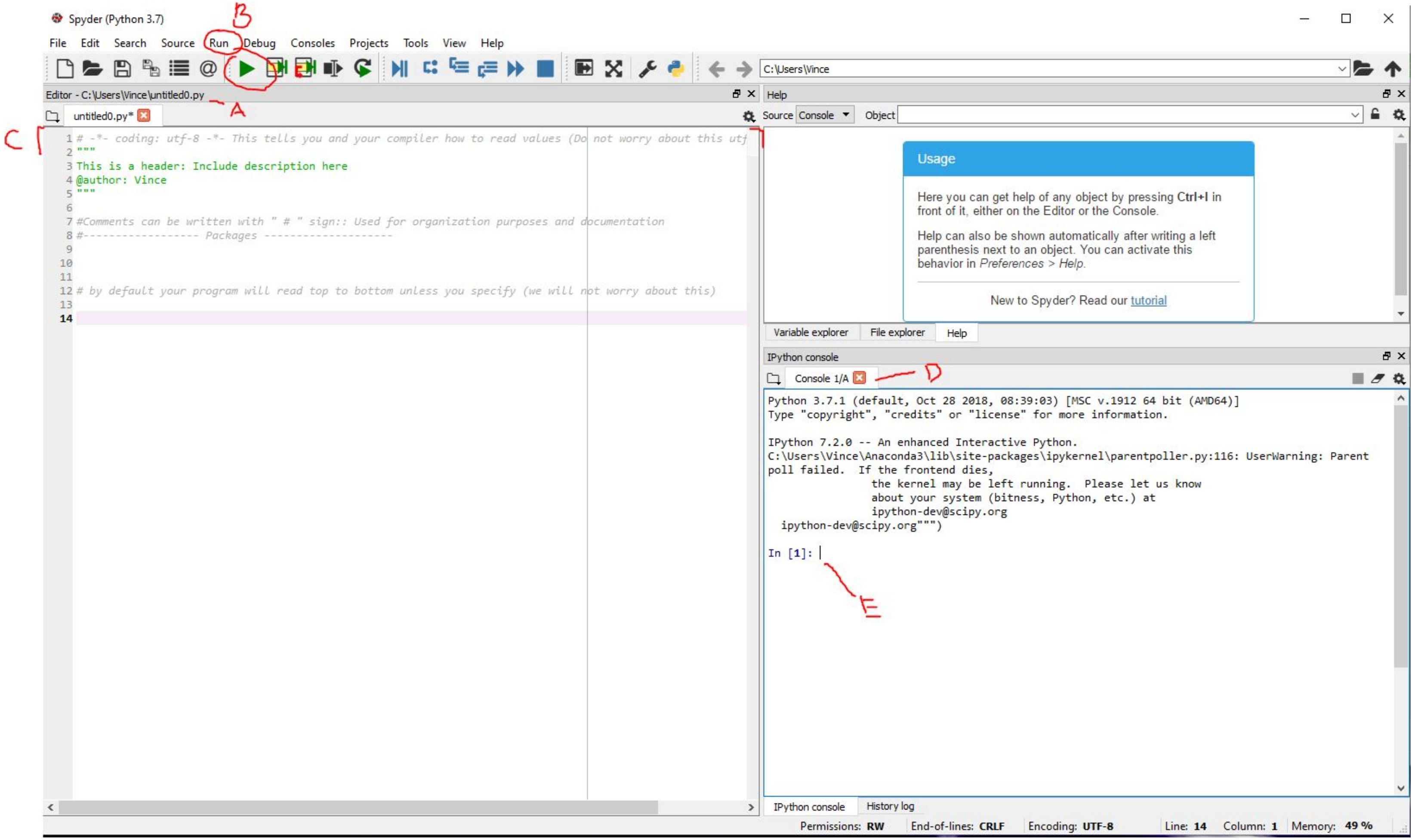
Component based data mining Framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Make sure you have Spyder (version 3.0+) and RStudio.
[3.0 ensures you’re running Python3 and not Python2]

If you installed the default settings for Anaconda it will come with all the packages we will be going over.

Download link: https://www.anaconda.com/distribution/
[Make sure it is downloaded for your operating system]



```
# -*- coding: utf-8 -*- This tells you and your compiler how to read values (Do not worry about this utf
"""
This is a header: Include description here
@author: Vince
"""

#Comments can be written with " # " sign:: Used for organization purposes and documentation
#------------------ Packages --------------------



# by default your program will read top to bottom unless you specify (we will not worry about this)

```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.7.1 (default, Oct 28 2018, 08:39:03) [MSC v.1912 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.2.0 -- An enhanced Interactive Python.
C:\Users\Vince\Anaconda3\lib\site-packages\ipykernel\parentpoller.py:116: UserWarning: Parent
poll failed.  If the frontend dies,
                the kernel may be left running.  Please let us know
                about your system (bitness, Python, etc.) at
                ipython-dev@scipy.org
  ipython-dev@scipy.org""")

In [1]: 
```

Permissions: **RW** End-of-lines: **CRLF** Encoding: **UTF-8** Line: **14** Column: **1** Memory: **49 %**

(Spyder IDE):
A - File location; B - Run program (quick run); C - Code editor; D - Console; E - Command line (similar to Jupiter Notebook)

The directory matters! It tells your program where to find your resources!

Crash Course: Coding

Supplemental material is widely available online:
https://www.learnpython.org/
https://www.youtube.com/results?search_query=python

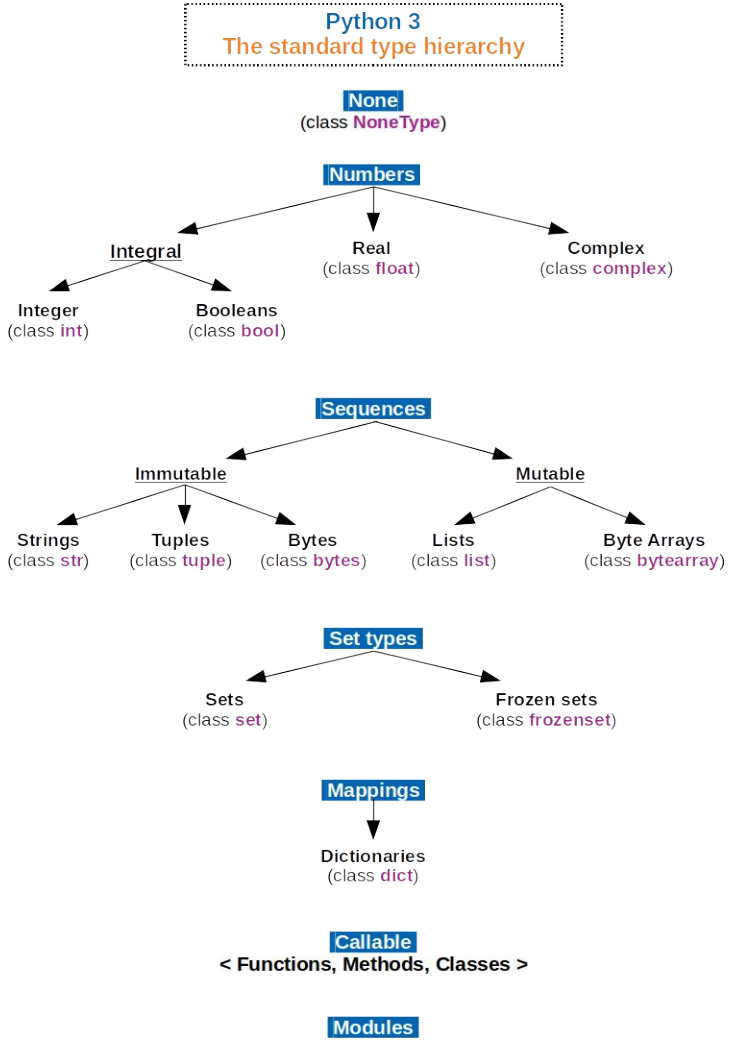
Fundamentals: Data types and structures

- Data type - a definition for the computer on how to interpret your data.
  - Ex: The number 1 can be interpreted as follows:
    - Integer (int) 1 [used for whole number calculations (will truncate)
    - Float/Double 1.0 [used for decimal or “floating point” calculations]
    - Char “1” [represents the character 1 - cannot perform mathematical operations]
    - Boolean 1 - boolean is a T/F value where a non-zero number implies T

Fundamentals: Data types and structures

- Data Structure - organization/storage/management format for data values/types.
  - Ex: Lists/arrays store sequences of data types
    - list = [1, 2, 3, 4]
    - Matrices
    - A string is a list of characters: “abcd”
    - Trees and graphs
    - You can even make your own

Cheat Sheet : Data types and structures



Python 3
The standard type hierarchy

None
(class NoneType)

Numbers

Integral

Integer
(class int)

Booleans
(class bool)

Real
(class float)

Complex
(class complex)

Sequences

Immutable

Strings
(class str)

Tuples
(class tuple)

Bytes
(class bytes)

Mutable

Lists
(class list)

Byte Arrays
(class bytearray)

Set types

Sets
(class set)

Frozen sets
(class frozenset)

Mappings

Dictionaries
(class dict)

Callable
< Functions, Methods, Classes >

Modules

Immutable = cannot edit

- Good for data you do not want to accidently modify

Mutable = can edit

- Good for when you want to modify/ perform calculations

Character (char) is often referred to as byte since they are essentially the same size and share the same representation

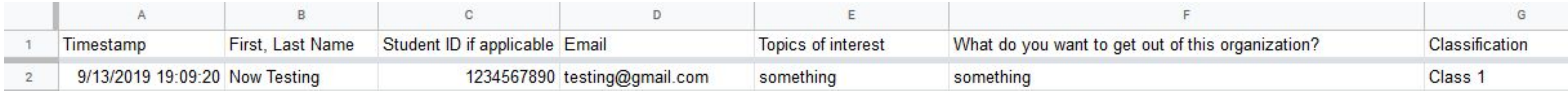
In Python if you do not declare a datatype with your variable it will not take on any attributes until declaration is needed (basically it's smart enough to figure out what you mean depending on what you need it to do [most of the time].)

If you want to read more about Data types/structures: (there's quite a lot)

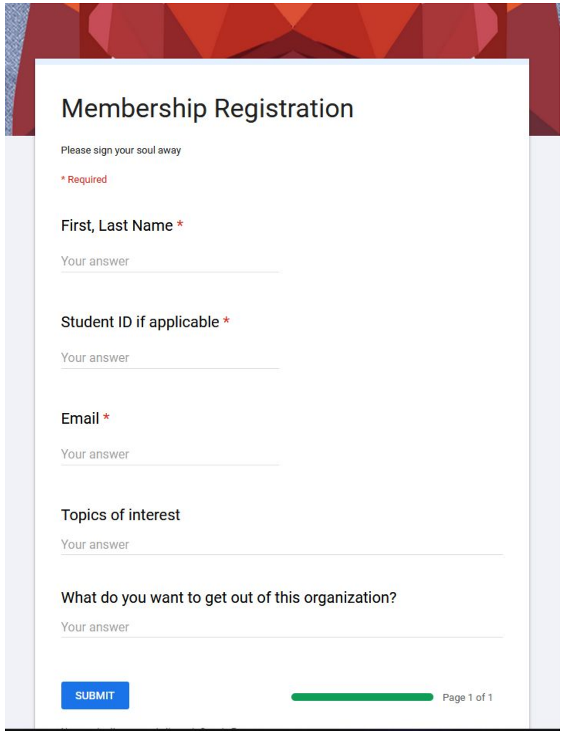
https://en.wikipedia.org/wiki/Data_type
https://en.wikipedia.org/wiki/Data_structure

(Credit):
https://commons.wikimedia.org/w/index.php?curid=74062464

Data type Analysis: What should I use?



	A	B	C	D	E	F	G
1	Timestamp	First, Last Name	Student ID if applicable	Email	Topics of interest	What do you want to get out of this organization?	Classification
2	9/13/2019 19:09:20	Now Testing	1234567890	testing@gmail.com	something	something	Class 1



Membership Registration

Please sign your soul away

* Required

First, Last Name *

Your answer

Student ID if applicable *

Your answer

Email *

Your answer

Topics of interest

Your answer

What do you want to get out of this organization?

Your answer

We can clearly see this registration form documents a multiple different data types.

Time is generally a continuous data type. It would make sense that this can be converted into a float or double.

Names are a collection of characters. A person’s name would be best stored in a string.

Student ID is a whole number. We should store this as an int.

Wordy responses would generally be strings or list of strings.

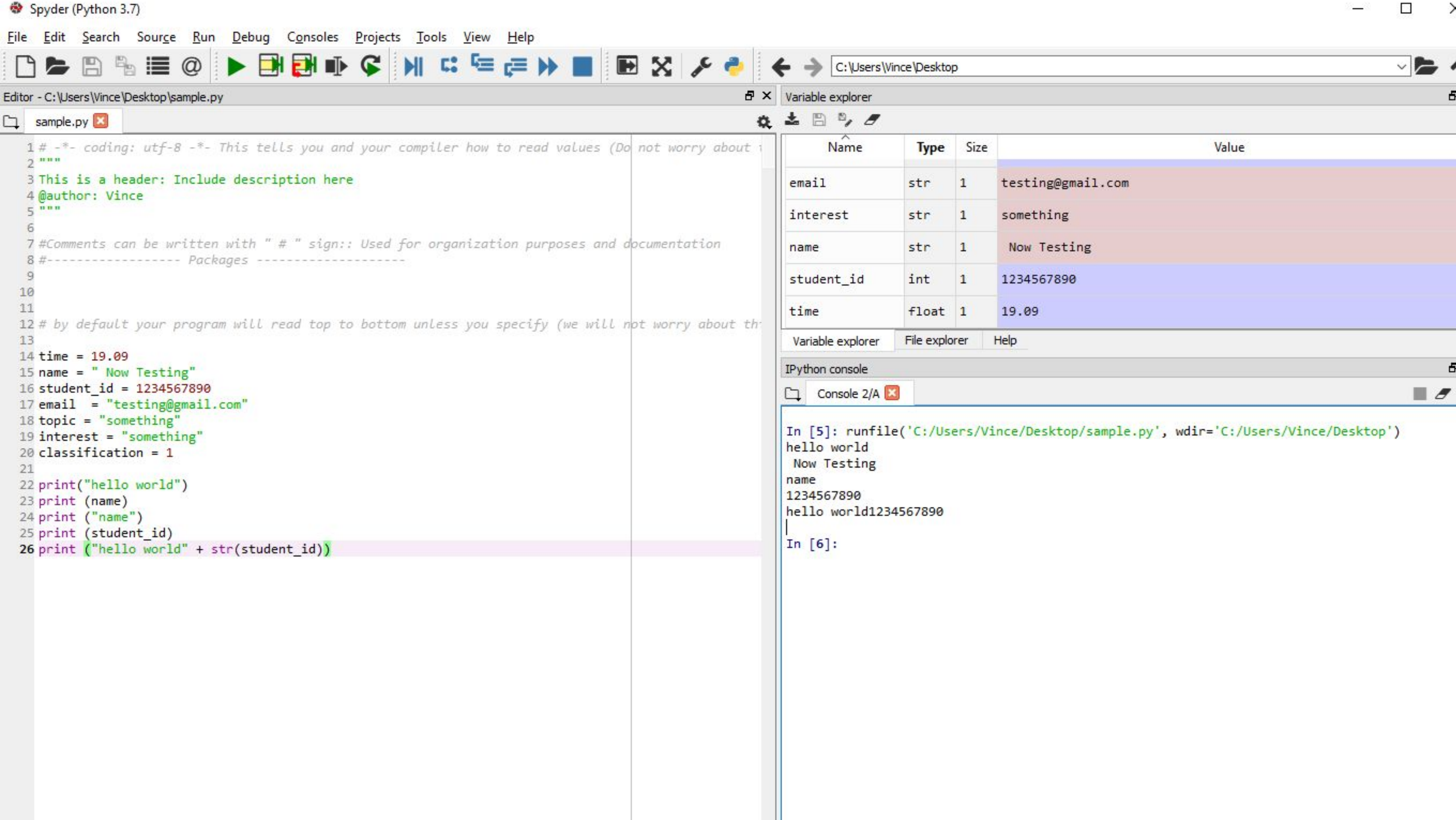
Classification (in supervised data they will be assigned a certain class) these classes will generally be converted to an integer but can be stored as a string.

Shameless plug: If you haven’t registered tinyurl.com/dsiregistration

Interactive Example

- Variable Declaration
  - Casting data types

- Print and display
  - print(_variable_)
  - print(“hello world”)
  - print() will always display the content inside the parentheses as a string.
    - use string concatenation and casting to link multiple prints.
- Lists and Indices



```python
# -*- coding: utf-8 -*- This tells you and your compiler how to read values (Do not worry about t
"""
This is a header: Include description here
@author: Vince
"""

#Comments can be written with " # " sign:: Used for organization purposes and documentation
#----------------- Packages -------------------



# by default your program will read top to bottom unless you specify (we will not worry about th

time = 19.09
name = " Now Testing"
student_id = 1234567890
email  = "testing@gmail.com"
topic = "something"
interest = "something"
classification = 1

print("hello world")
print (name)
print ("name")
print (student_id)
print ("hello world" + str(student_id))
```

Name	Type	Size	Value
email	str	1	testing@gmail.com
interest	str	1	something
name	str	1	Now Testing
student_id	int	1	1234567890
time	float	1	19.09

```
In [5]: runfile('C:/Users/Vince/Desktop/sample.py', wdir='C:/Users/Vince/Desktop')
hello world
 Now Testing
name
1234567890
hello world1234567890

In [6]:
```

Bread and Butter: What are packages?

- Code someone else has written that you can load and use.
  - If you need to do something, someone probably has already created a package to do it.
- Packages streamline your workload
  - Don’t reinvent the wheel
  - If you’re new to programming, it will probably be computationally more efficient than if you wrote the code for it.
- **IF YOU DON’T KNOW: READ DOCUMENTATION**
  - If you didn’t code it, make sure you follow instructions on how the code works. It may not work exactly how you think it would.

Cookie Cutter: Packages to use

Numpy - Optimizer for matrices work

https://docs.scipy.org/doc/numpy/user/index.html

SKLearn - Data Science algorithms

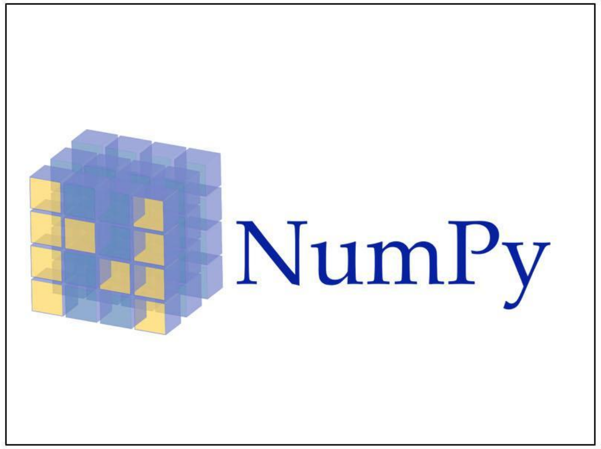
https://scikit-learn.org/stable/documentation.html

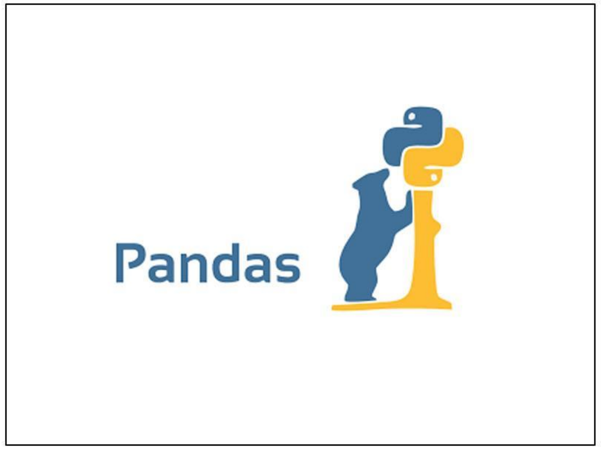
Pandas - Dataframe manipulation

https://pandas.pydata.org/pandas-docs/version/0.25/

MatPlotLib - Graphs and representation

https://matplotlib.org/users/index.html





Interactive Pandas Demonstration

› Import packages

› Function calling

- Sorting
- Display

› Loading Data

- have simple_retail.csv downloaded

For demonstration purposes the dataset available on the Github is truncated. For the full dataset you can find it here at:

Intro to Machine Learning

What Machine Learning really is

- Formally it's utilizing computation power to approximate models.

- Informally its voodoo magic but we'll call it implicit modeling for job security

- Subcategory of AI with a large application to Data Science
  - Find patterns in data (data mining)
  - Model data to predict
  - SCALABLE

Applications

- Image processing and analysis
  - Medical imaging
  - Facial profiling
- Pattern Detection
  - Find relationships and model unknown behavior
  - For business decisions
- Classification and filtering
  - Sort through mounds of data and categorize
  - Automate machine jobs

Applications: Personal Anecdote

- Disease detection in medical images
- Generating and Disseminating fake ~~news~~ images
- Modeled drinking water purity for carcinogens

Intro to Clustering

Dealing with Unsupervised Data

- If we are not given any information about our data
  - Use clustering

- How can we find patterns on something we don't know?
  - Sort your data into groups
  - Then figure out the relationships in those group

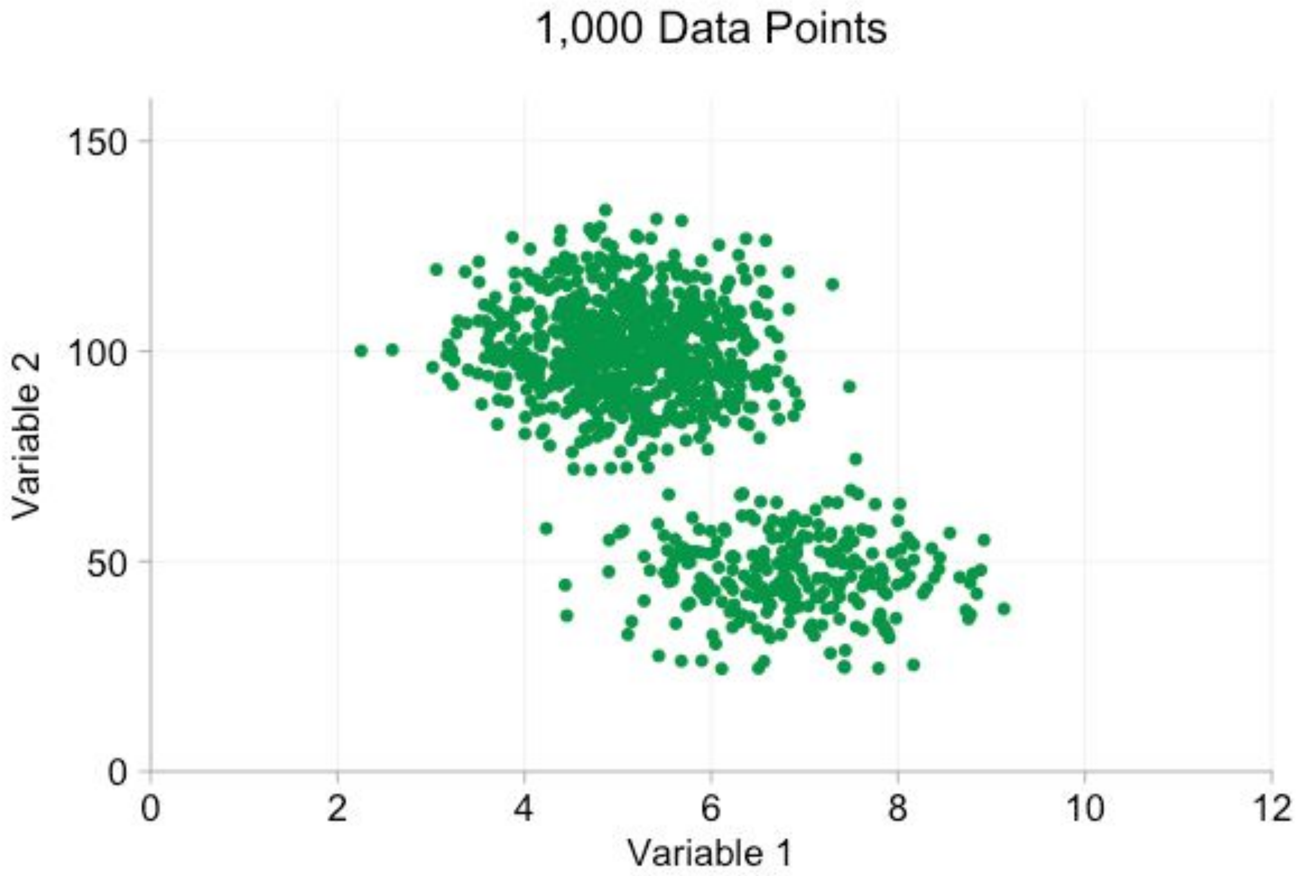
Disclaimer: works with any dimensions! It's just hard to visualize after 3D

Clustering: A not realistic example

- Say a random guy named Mark Zuckerberg somehow acquired information about a large group of people
  - His data consisted of a lot of things but he is only interested in:
  - Location, Hobbies

- Without knowing anything else, how can Mr. Zuckerberg find out the correlations between these attributes if any?
  - Plot the data
  - Then figure out how many possible distinct groups

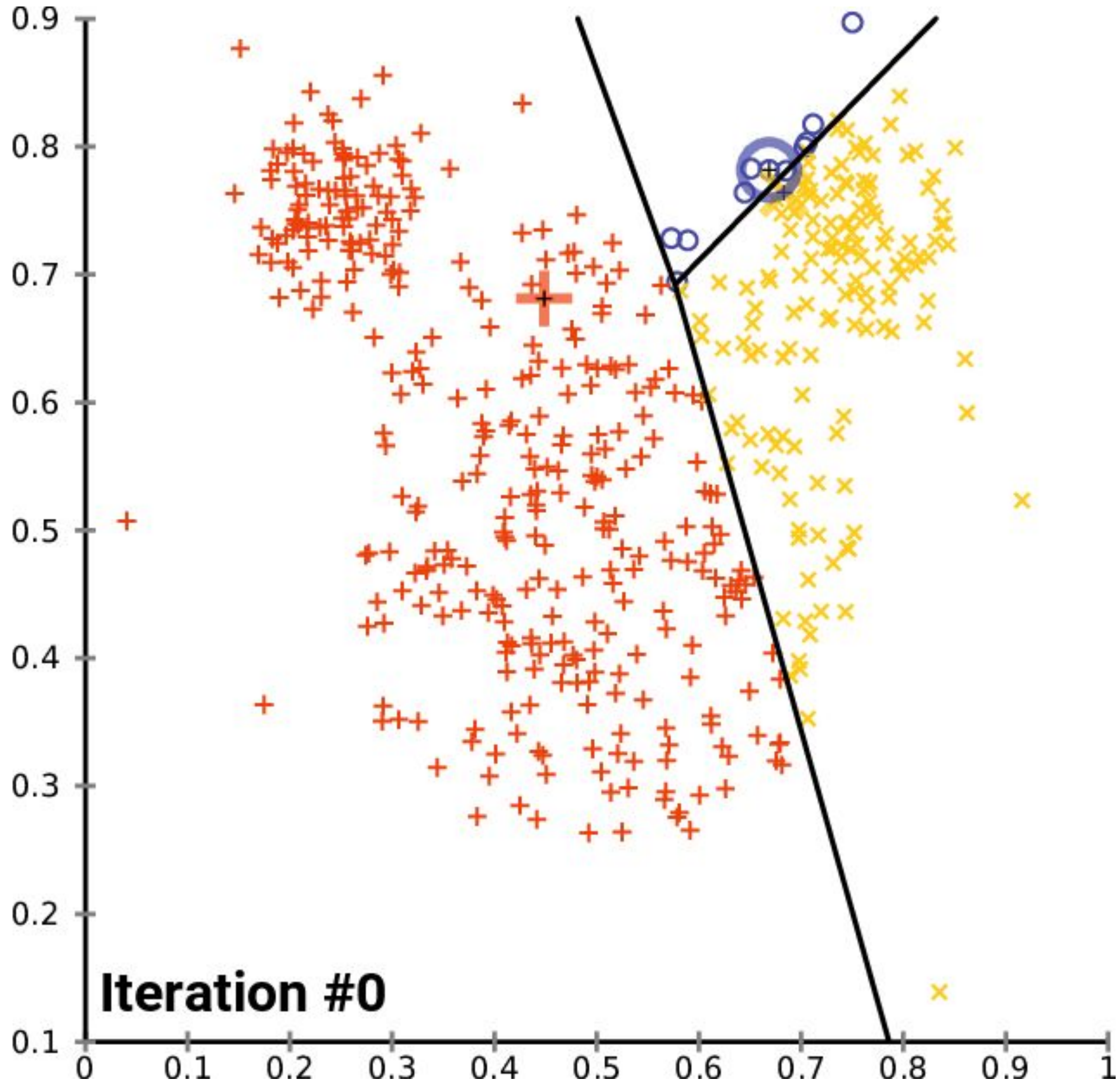
Clustering: A not realistic example



In reality data NEVER looks this defined. We can estimate about 2 groups or “clusters” is a reasonable representation. Let’s just say for simplicity, Mr. Zuckerberg finds these two clusters. He can separate his data and find the correlation in those two groups. Maybe the common factor in these two clusters were age.

K-Means Clustering in a Nutshell

- Clustering Algorithm.
- User defines K-number of clusters to look for
- Machine will randomly select K-centroids (centers)
  - All data points will be assigned the class of the nearest center
  - Machine will update the centers that minimize the maximum distance of worst data point.
  - Update will stop after a defined number of iterations or convergence


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Iteration #0

Interactive K-Means Clustering

- Run k-means.py
- Function calls
- Plotting

Q&A / AMA

- Potential Future Topics:
  - Associate Rule Mining (Data Mining)
  - Other clustering algorithms
  - Neural Networks
  - Distributed File Systems (DFS) and Cloud Computing
- Projects and Competition