Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

PYTHON PROGRAMMING FOR DATA SCIENCE – PART 2

MASSIMILIANO IZZO & NICHOLAS DAY

LECTURE 1
INTRODUCTION TO THE COURSE
INTRODUCTION TO DATA SCIENCE

Welcome to “Python Programming for Data Science”

- Developed for the Department of Continuing Education:

https://gitlab.com/data-science-course/pp4ds-pt2-tt2022

- Tutor: Massimiliano Izzo
- Live Sessions: Fridays 5-6pm on Microsoft Teams
- Q&A: Using online workspace on Canvas and/or Slack (pp4ds-ox.slack.com)
- Email: massimiliano.izzo@conted.ox.ac.uk

Week 1

- Introduction to the course
- What is Data Science? (the short of it)
- Introduction to Machine Learning
- Exercise: Matrices and a Linear Regression Example

Announcements

https://gitlab.com/data-science-course/pp4ds-pt2-tt2022

About your tutor

- Originally from Genoa, Italy
- Biomedical Engineering degrees from the University of Genoa
- Worked for 5 years as RSE at the University of Oxford
- Software Engineer at the Oxford-based AI company Mind Foundry
- Contacts
  - email: massimiliano.izzo@conted.ox.ac.uk
  - Slack: using the pp4ds-ox.slack.com workspace

UNIVERSITÀ DEGLI STUDI DI GENOVA

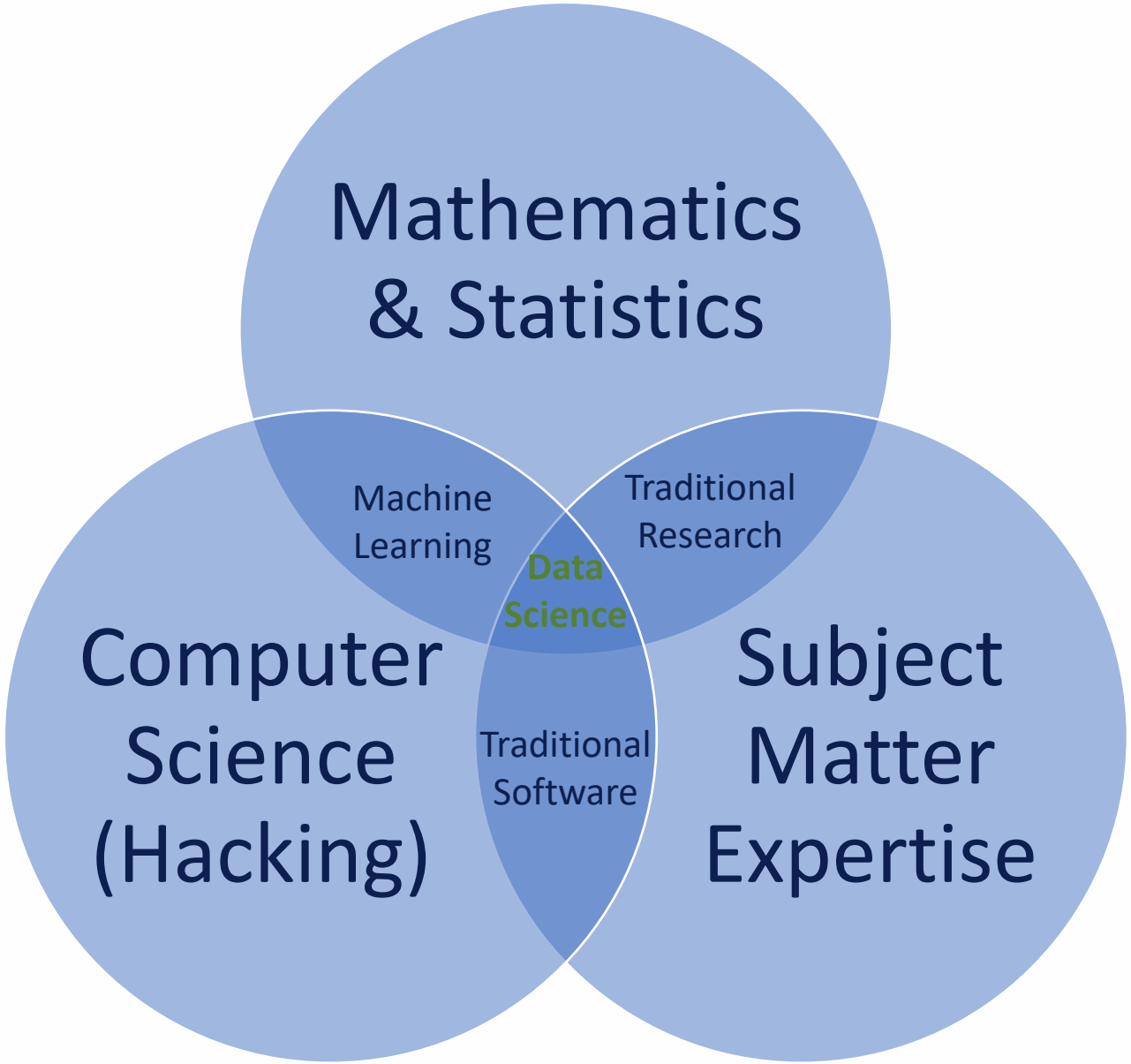
DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

mind FOUNDRY

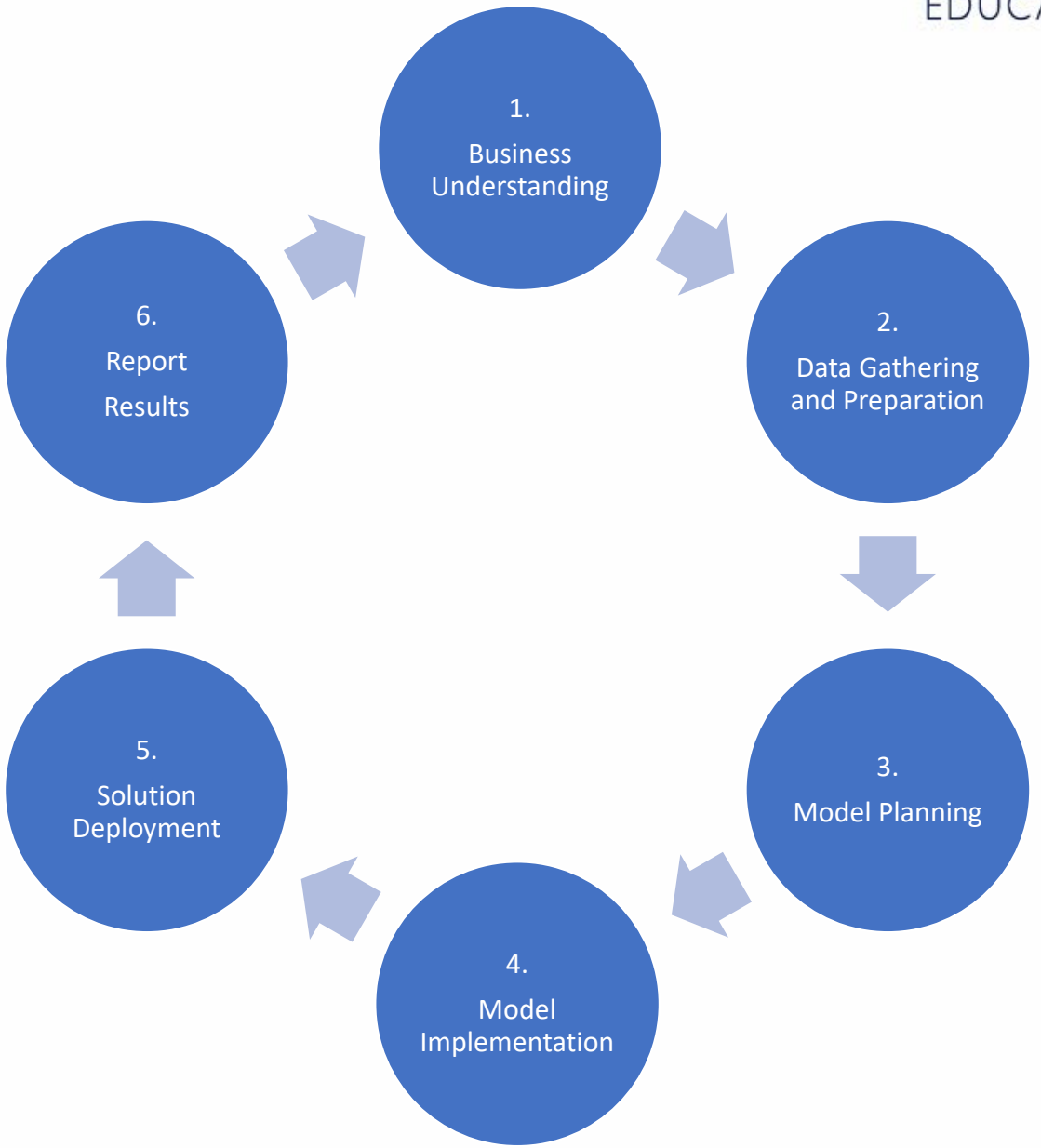
What is Data Science?

What is Data Science?

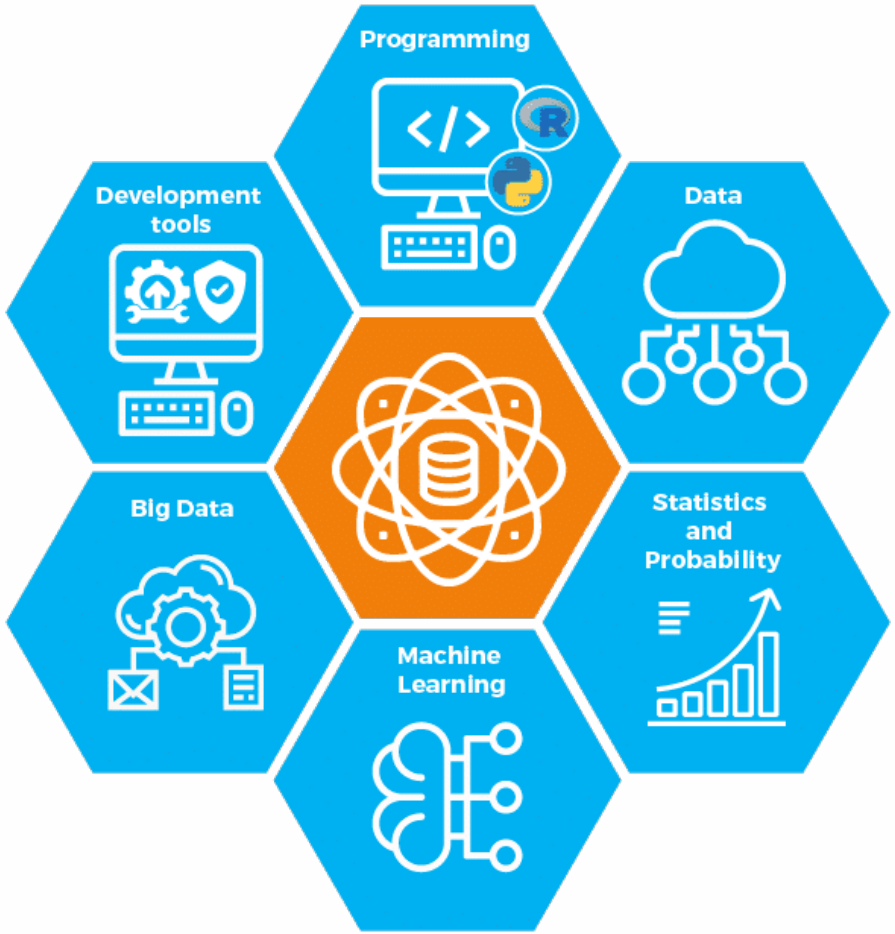
- A collection of methods and processes to explore real world problems with data
- These may include things like:
  - Collecting data
  - Cleaning the data
  - Storing and organizing the data
  - Analyzing and extracting insights from the data
- Ultimately, the point is to use data to inform decision making


Mathematics & Statistics
Machine Learning
Traditional Research
Data Science
Computer Science (Hacking)
Traditional Software
Subject Matter Expertise

Data Science Life Cycle

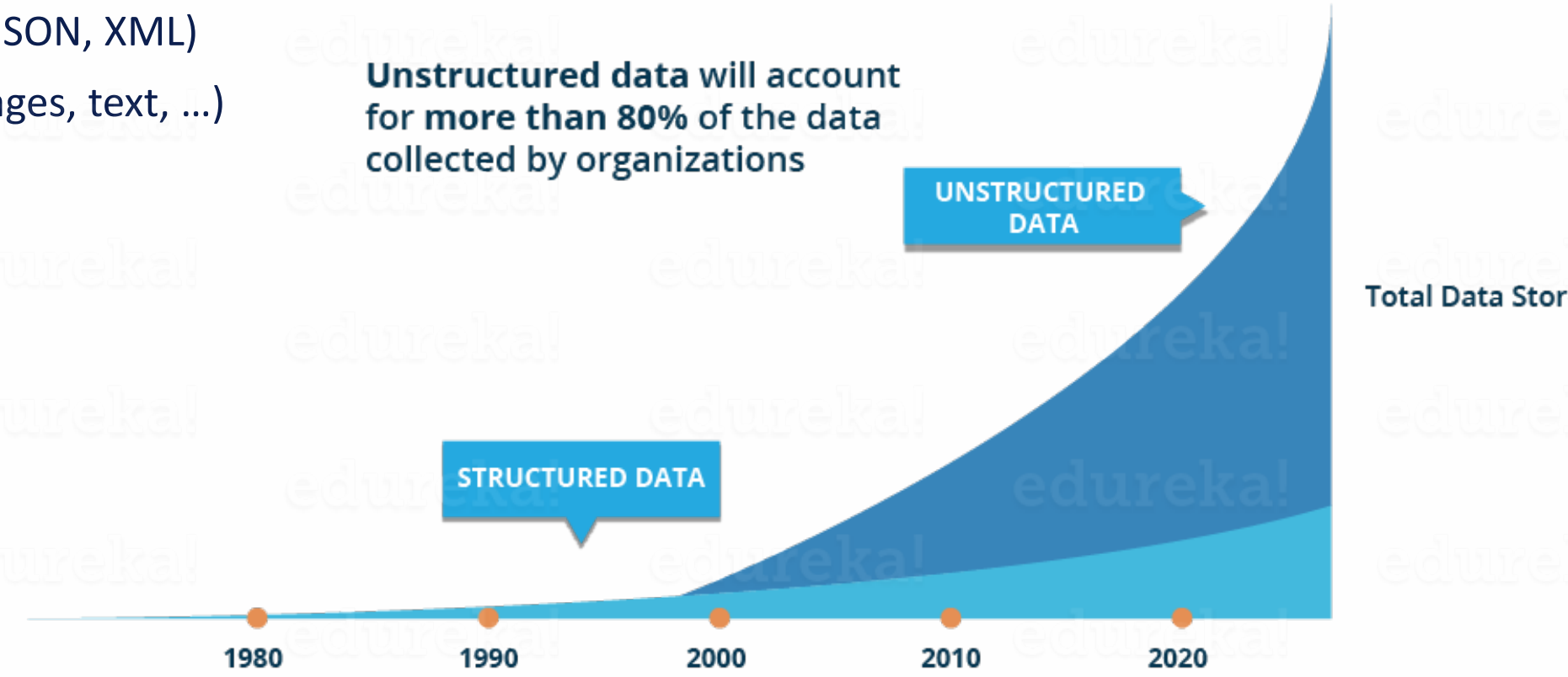


Data Science Components (https://intellipaat.com)



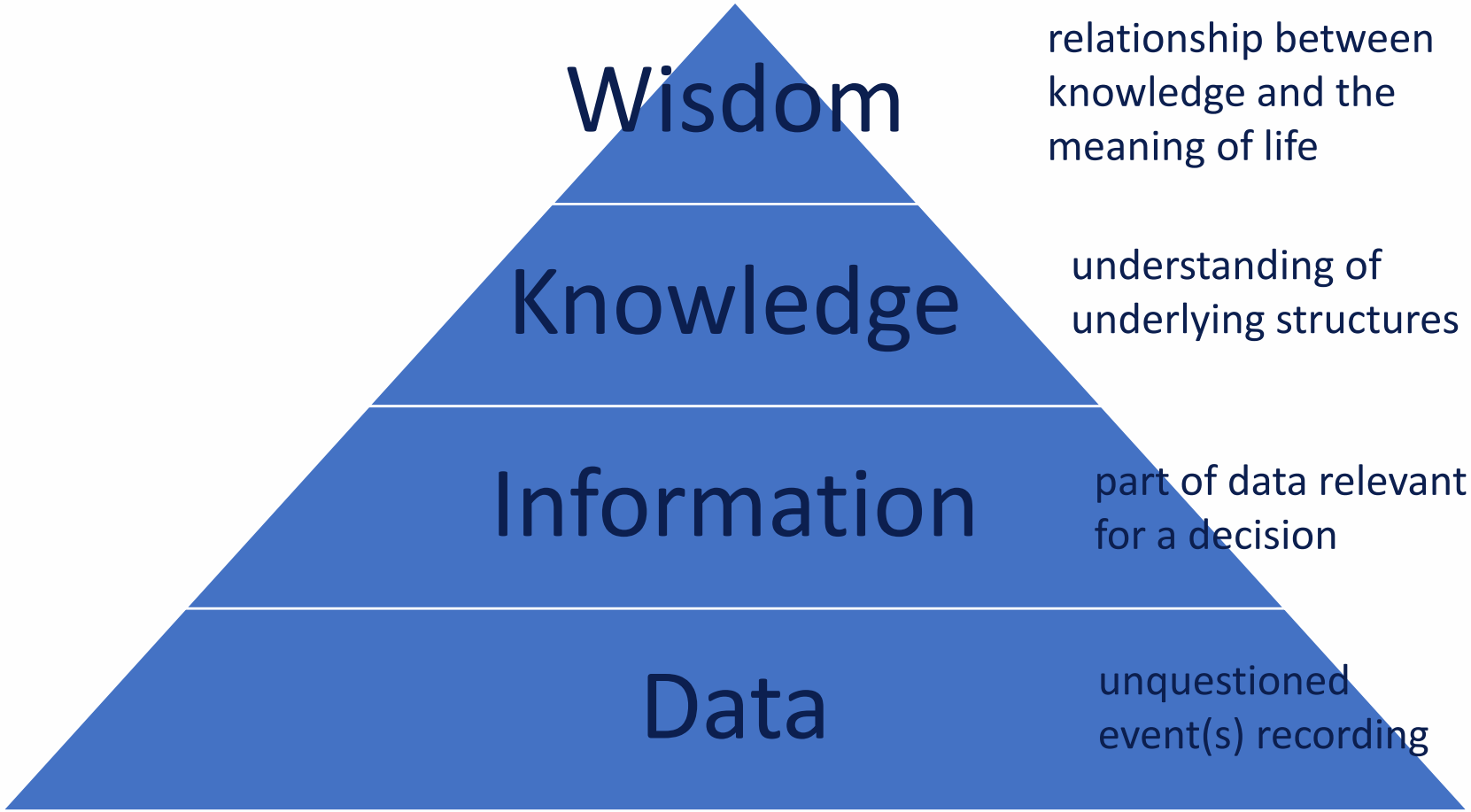
Data

- Structured (tabular)
- Semi-structured (JSON, XML)
- Unstructured (images, text, …)



https://www.edureka.co/blog/what-is-data-science/

DIKW (epistemological) pyramid

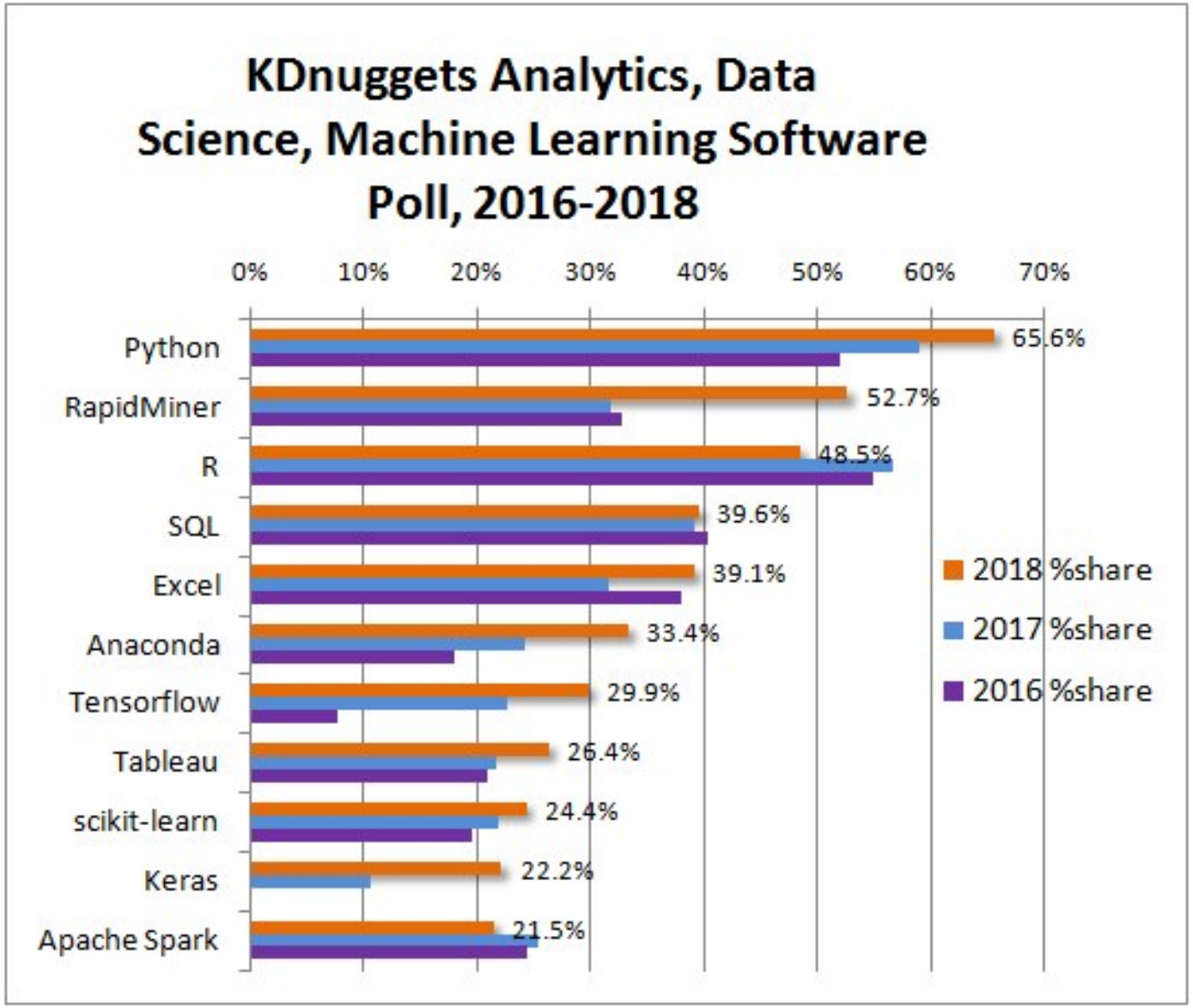


https://www.adizes.com/articles/cval-where-wisdom.pdf

Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD



Why Python?

- Popular in data science and also for general software development
- Mastering the language basics is essential
- Good news: this course is about the basics!

Why Python?

- Easy to write code but can handle complex mathematical processing
- Requires fewer lines than C++, Java, or R to achieve similar operations
- No curly braces, indentation (4 spaces or tabs) is compulsory
- Code can be executed in batch, or interactively
- Free and open source ecosystem
- Relatively high performance

Python libraries for Data Science

- https://activewizards.com/blog/top-15-libraries-for-data-science-in-python/

Numeric computation	NumPy
Scientific computation	SciPy library
Visualization	Matplotlib, Seaborn, Bokeh, Ploty
Machine Learning	Scikit-learn
Deep Learning	Tensorflow, Keras, Jax
Natural Language Processing	NLTK, Gensim
Others	Scrapy, Statsmodels, Spyder, Jupyter

Components of P4DS

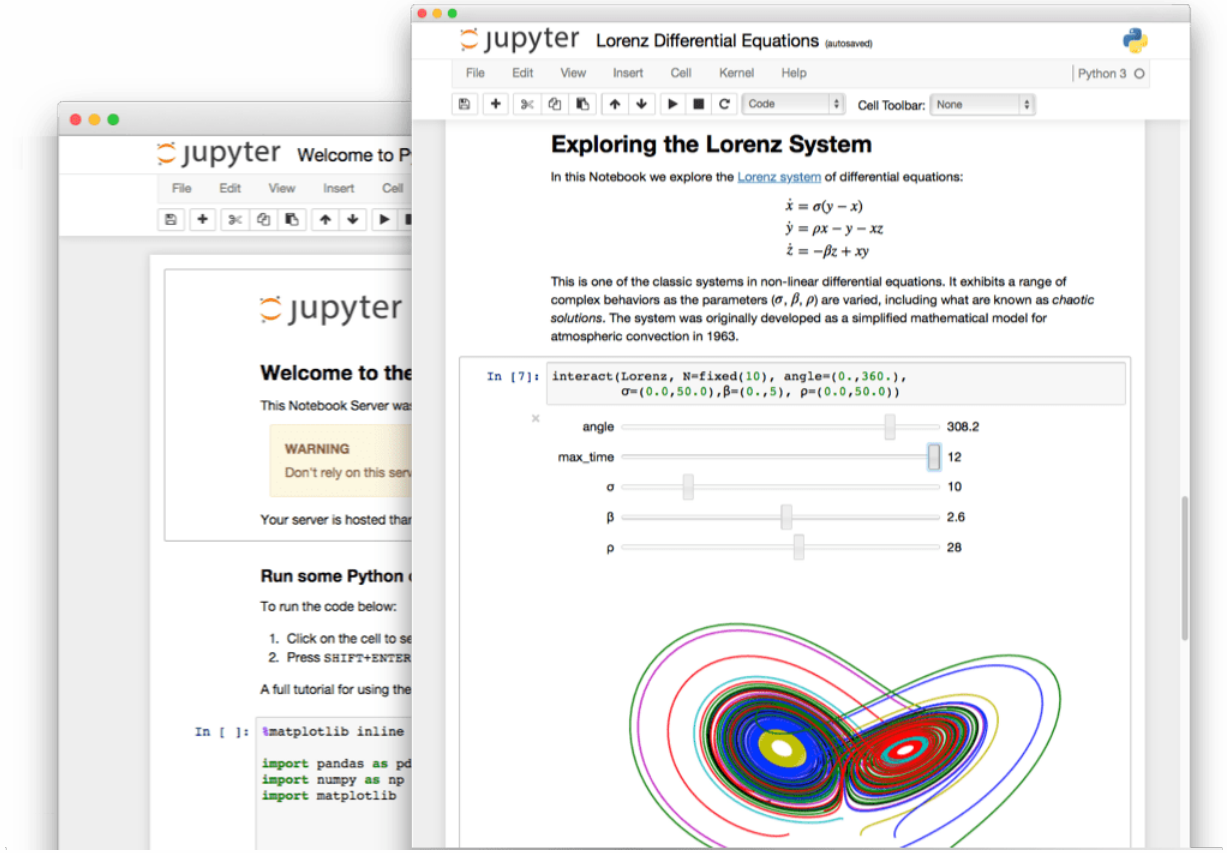
- Lectures
- Practical programming exercises, using Python
- Class discussions, off and online
- A record of attendance is kept
- Summative Assignment: a portfolio of exercises
- Week-to-week class evaluation forms

Collaboration

- **Asking questions is encouraged!**
  - Discuss all questions between you
  - Help each other out in the practical exercises
  - It's also OK to ask Google (or any other Internet search engine)
- **Limits**
  - When you submit your Final Assignment for assessment, it must be an **individual piece of work**

Jupyter Notebooks

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.



Source: https://jupyter.org/

Working with Jupyter: Online Option

- Go to our course web page on Gitlab.com (**https://gitlab.com/data-science-course/p4ds-ht2020**)
- Start Jupyter notebook clicking on the Binder badge next to the exercise name

Working with Jupyter: Offline Option

- Open GitBash (or a Windows Terminal)
- From your Home Directory run this command (all in one line): **git clone https://gitlab.com/data-science-course/p4ds-ht2020**

- Start on your machine Anaconda Navigator
- Start Jupyter notebook from the interface
- Open the notebook (guidance will be provided)

This week and beyond…

W1: Basic overview of Machine Learning. Linear Regression example.

W2: Overview of a data-science preprocessing pipeline.Data Inspection, Cleaning, and Pre-processing with scikit-learn

W3: Supervised Learning: regression models with scikit-learn

W4: Supervised Learning: classification models with scikit-learn

W5: Decision Trees. Ensemble Methods. The Perceptron.

W6: Deep Learning: Feed-forward Neural Networks. with keras and Tensorflow

W7: Natural Deep Learning: Convolutional Neural Networks (CNNs) for Image Processing. Recurrent Neural Networks (RNNs) for time series analysis.

W8: Dimensionality reduction and Unsupervised Learning with scikit-learn

W9: Natural Language Processing (NLP): an overview. Word embeddings. RNNs for NLP. Attention-based models (Transformers).

W10: Autoencoders and Generative Adversarial Networks (GANs). Explainability of Machine Learning Models.

Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

PYTHON PROGRAMMING FOR DATA SCIENCE – PART 2

MASSIMILIANO IZZO & NICHOLAS DAY

LECTURE 1
INTRODUCTION TO MACHINE LEARNING

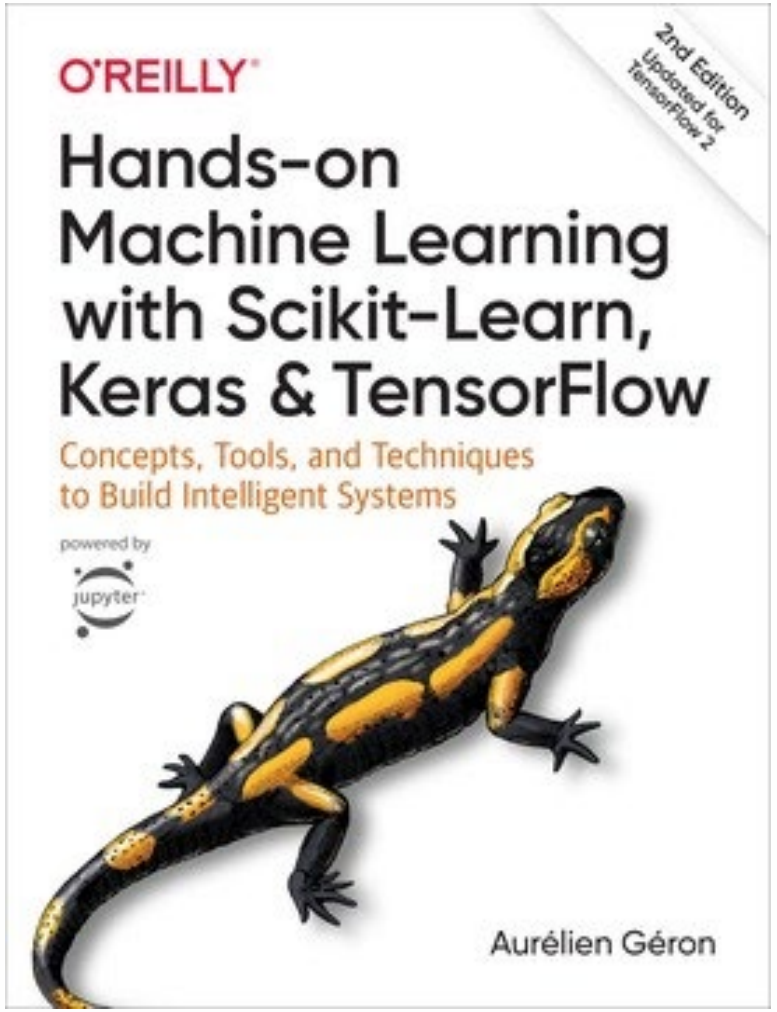
What is Machine Learning?

- “[ML is] the field of study that gives computers the ability to learn without being explicitly programmed.” (Arthur Samuel, 1959)
- “A computer program is said to learn from experience *E* with respect to some task *T* and some performance measure *P*, if its performance on *T*, as measured by *P*, improves with experience *E*.” (Tom Mitchell, 1997)

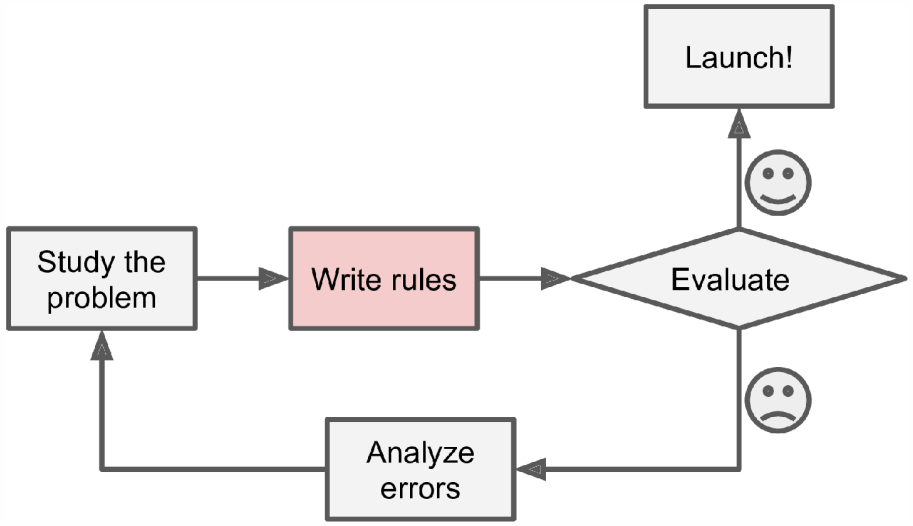
"Hands-on Machine Learning" Book

Aurélien Géron, ***Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems***
O'Reilly Media; 2 edition (October 15, 2019)
ISBN-10: 1492032646
ISBN-13: 978-1492032649

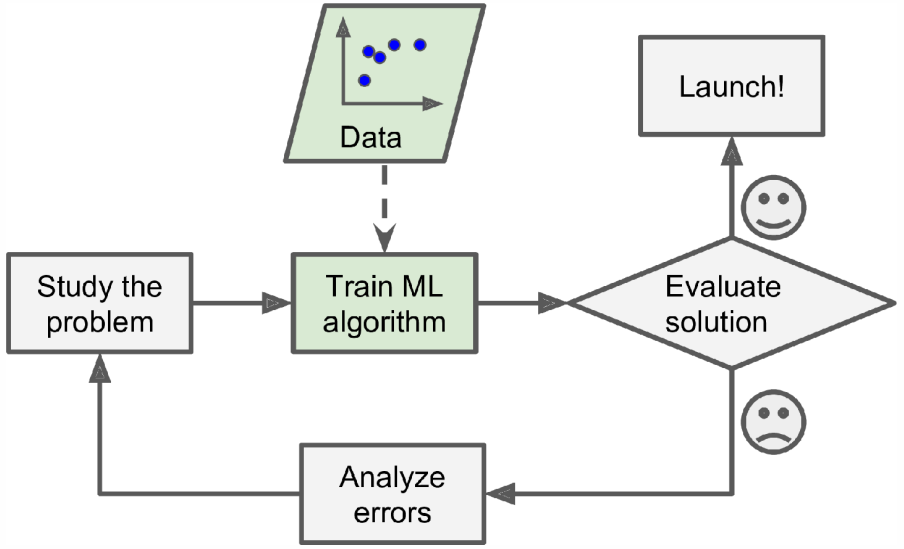
856 pages (it covers much more that what we will see in the first 6 weeks of the course)



The Traditional Approach



The Machine Learning Approach



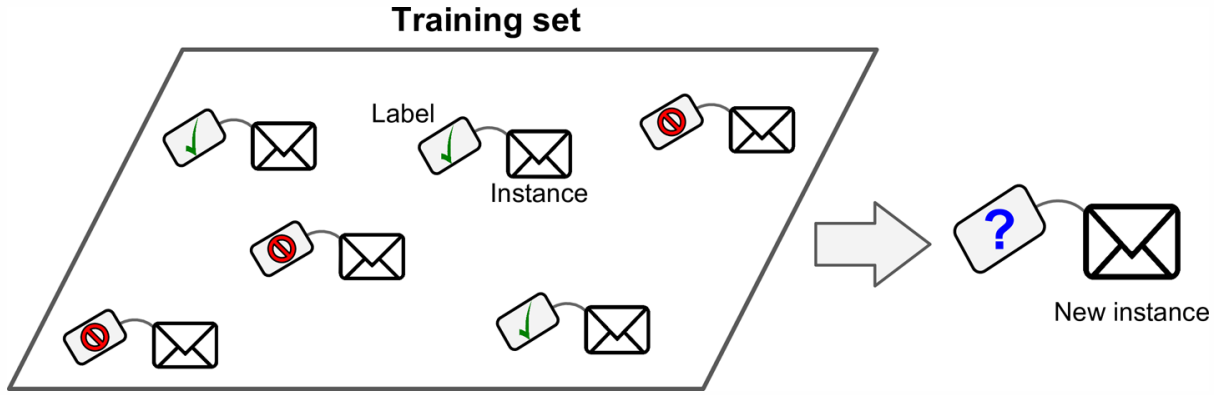
Source: Aurélien Géron, ***Hands-On Machine Learning***

Types of Machine Learning Systems

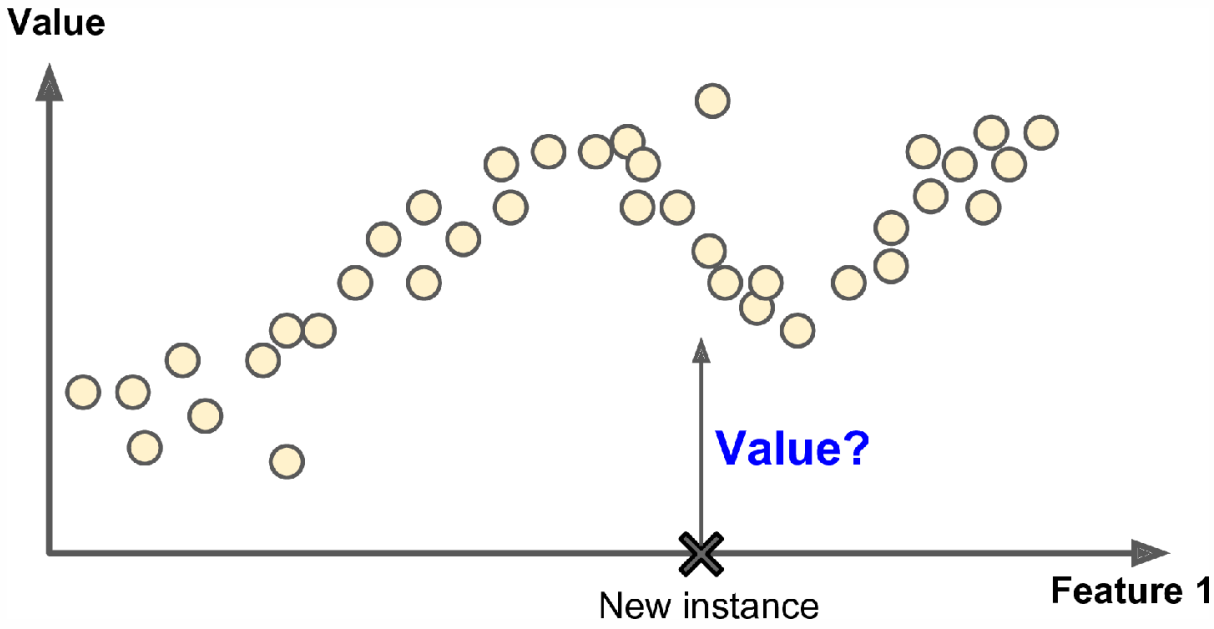
- Are they trained with human supervision? (Supervised, Unsupervised and Reinforcement Learning)
- Can they learn incrementally on the fly? (online versus batch learning)
- Can they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model (instance-based versus model-based learning)

Supervised Learning

Classification: to which class does an instance belong to? (e.g. is this email ham or spam?)



Regression: given a set of pairs (Xi, Yi) how can I fit them to a function y=f(X)?



Source: Aurélien Géron, ***Hands-On Machine Learning***

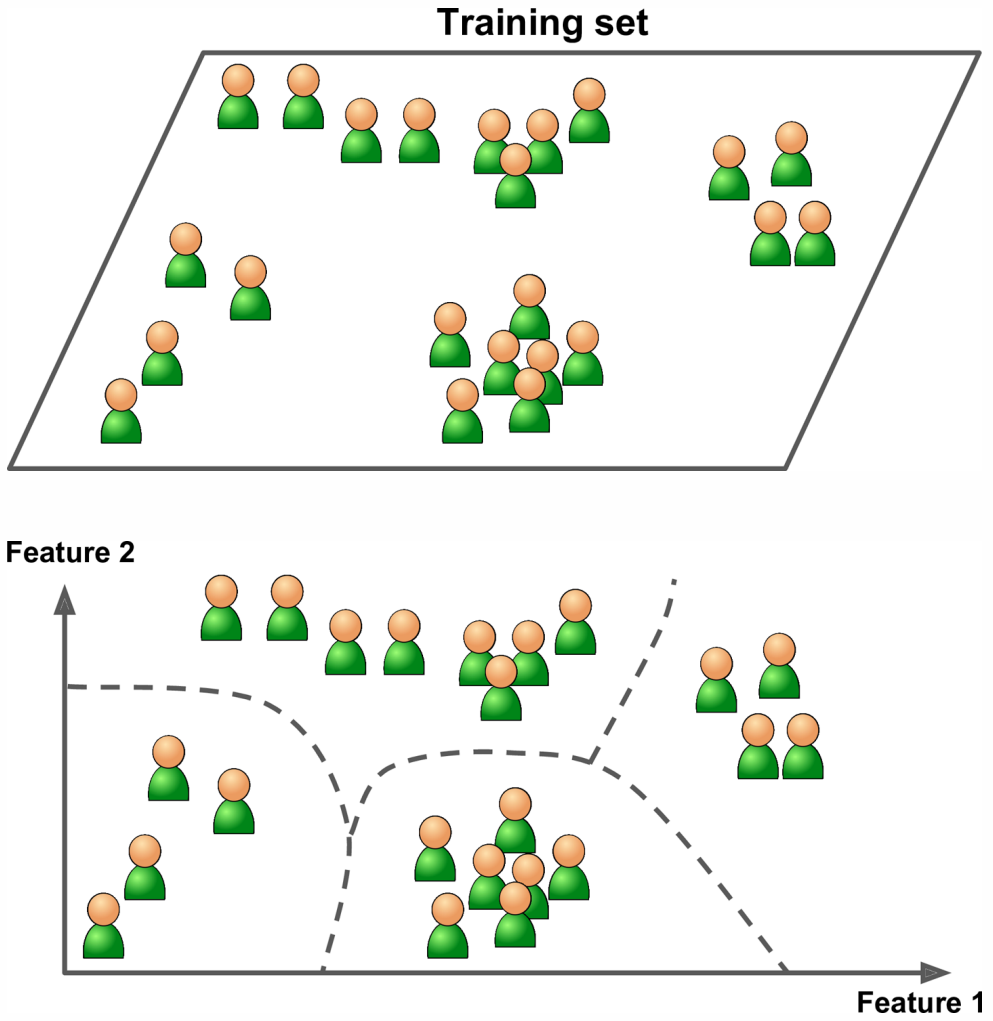
Supervised Learning: examples

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Naïve Bayes Classification
- Support Vector Machines (SVMs)
- Gaussian Processes
- Decision Trees and Random Forests
- Ensemble Methods
- Neural networks

Unsupervised Learning

Finding previously unknown patterns in data set without pre-existing labels.
Examples:

- Clustering (hierarchical clustering, K-means, DBSCAN, gaussian mixtures)
- Anomaly Detection
- Dimensionality Reduction (PCA, Locally Linear Embedding)
- Neural Networks (Autoencoders, Deep belief nets, Generative adversarial networks)

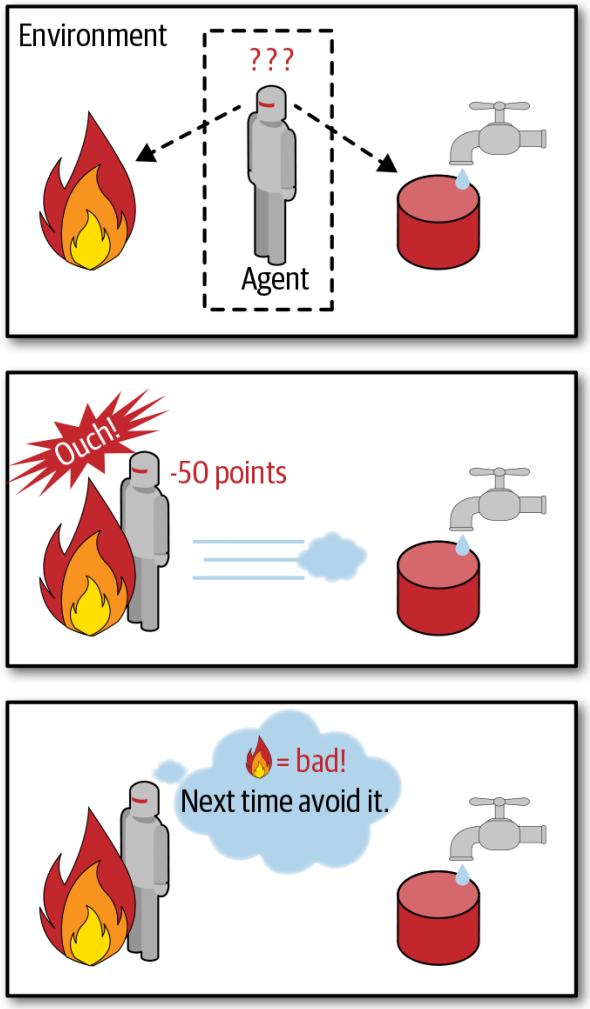


Source: Aurélien Géron, ***Hands-On Machine Learning***

Reinforcement Learning

An agent must take suitable action to maximize reward in a particular situation.

- Markov Decision Processes
- Q-Learning
  - Deep Q-Learning



1. Observe
2. Select action using policy
3. Action!
4. Get reward or penalty
5. Update policy (learning step)
6. Iterate until an optimal policy is found

Source: Aurélien Géron, ***Hands-On Machine Learning***

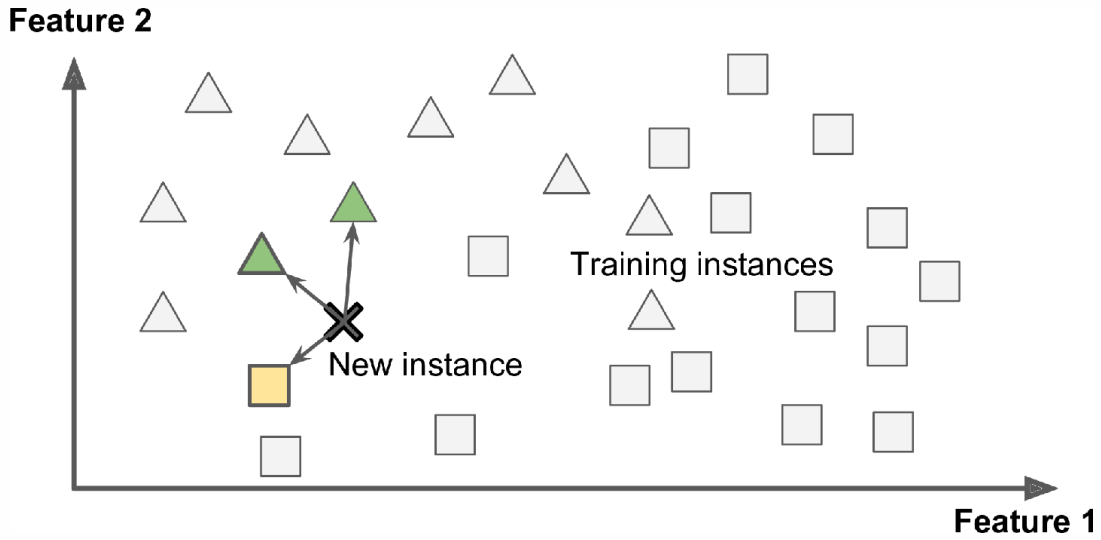
Other Machine Learning paradigms

- Besides the three main paradigms defined before (Supervised, Unsupervised, Reinforcement Learning) there are some approaches that fall somehow on the edges of them:
  - **Semi-supervised learning**: only a minority of the samples in the dataset have labels. Pseudo-labelling techniques must be used to predict the missing labels
  - **Self-supervised learning**: is in some sense a type of unsupervised learning as it follows the criteria that no labels were given. However, instead of finding high-level patterns for clustering, self-supervised learning attempts to still solve tasks that are traditionally targeted by supervised learning (e.g., missing word prediction in a text corpus) without any labels available.

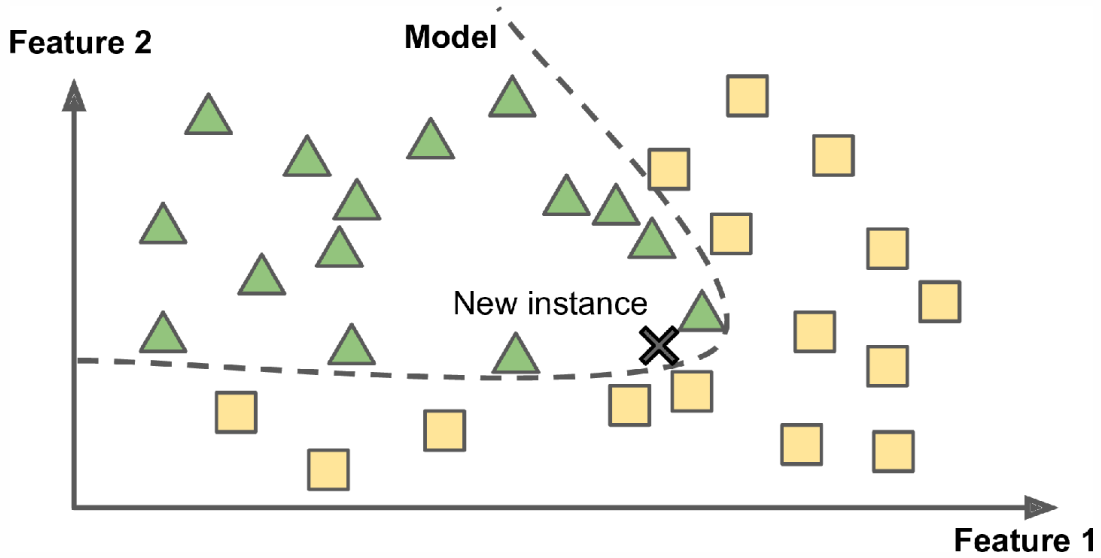
Batch vs Online Learning

- *batch learning:* system not capable of learning incrementally: it must be trained using all the available data => a lot of time and computing resources => typically done offline. (1) the system is trained, and (2) it is launched into production (no more learning). This is called *offline learning*.
- *online learning:* system trained incrementally receiving data instances sequentially, either individually or in small groups called *mini-batches*. Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives

Learn by Instance



Learn by model



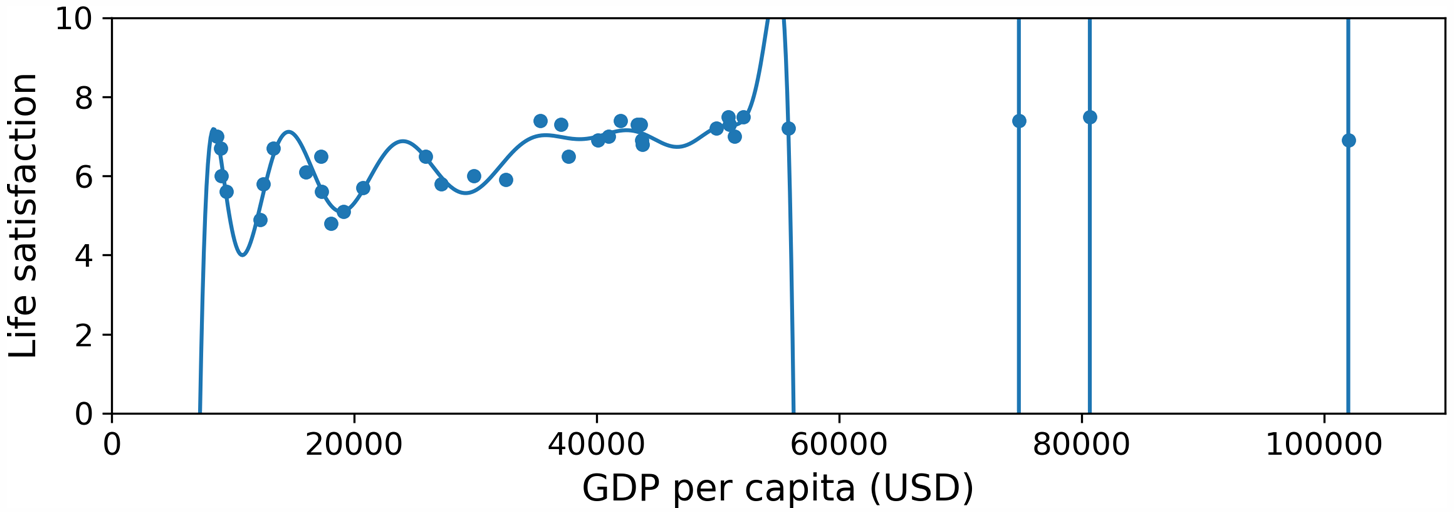
Source: Aurélien Géron, ***Hands-On Machine Learning***

The Unreasonable Effectiveness of Data

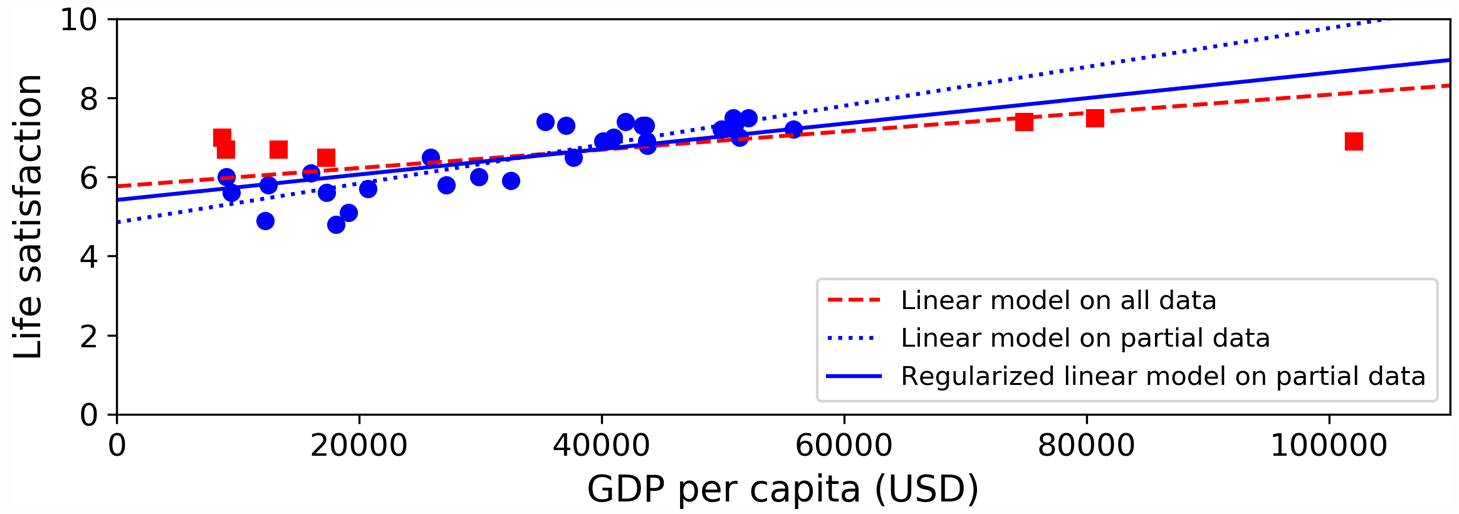
- Different Machine Learning algorithms, including fairly simple ones, performed almost identically on a complex problem if enough data was provided
- “these results suggest that we may want to reconsider the trade-off between spending time and money on algorithm development versus spending it on corpus development.”
  - Michele Banko and Eric Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France. Association for Computational Linguistics.
- However, small and medium-sized datasets are still fairly common => algorithms are still important

Overfitting and regularisation

the model performs well on the training data, but it does not generalize well.



Constraining a model to make it simpler and reduce the risk of overfitting is called *regularisation*.



Source: Aurélien Géron, ***Hands-On Machine Learning***

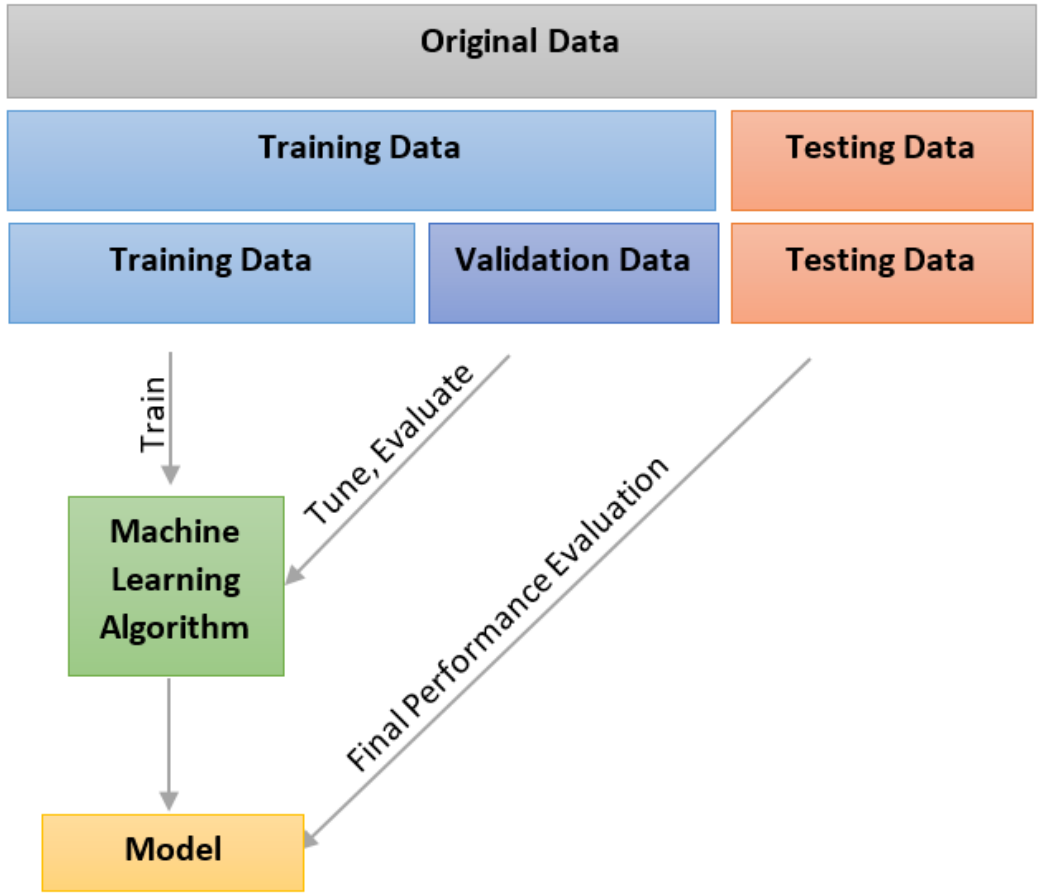
Training, Testing and Validation

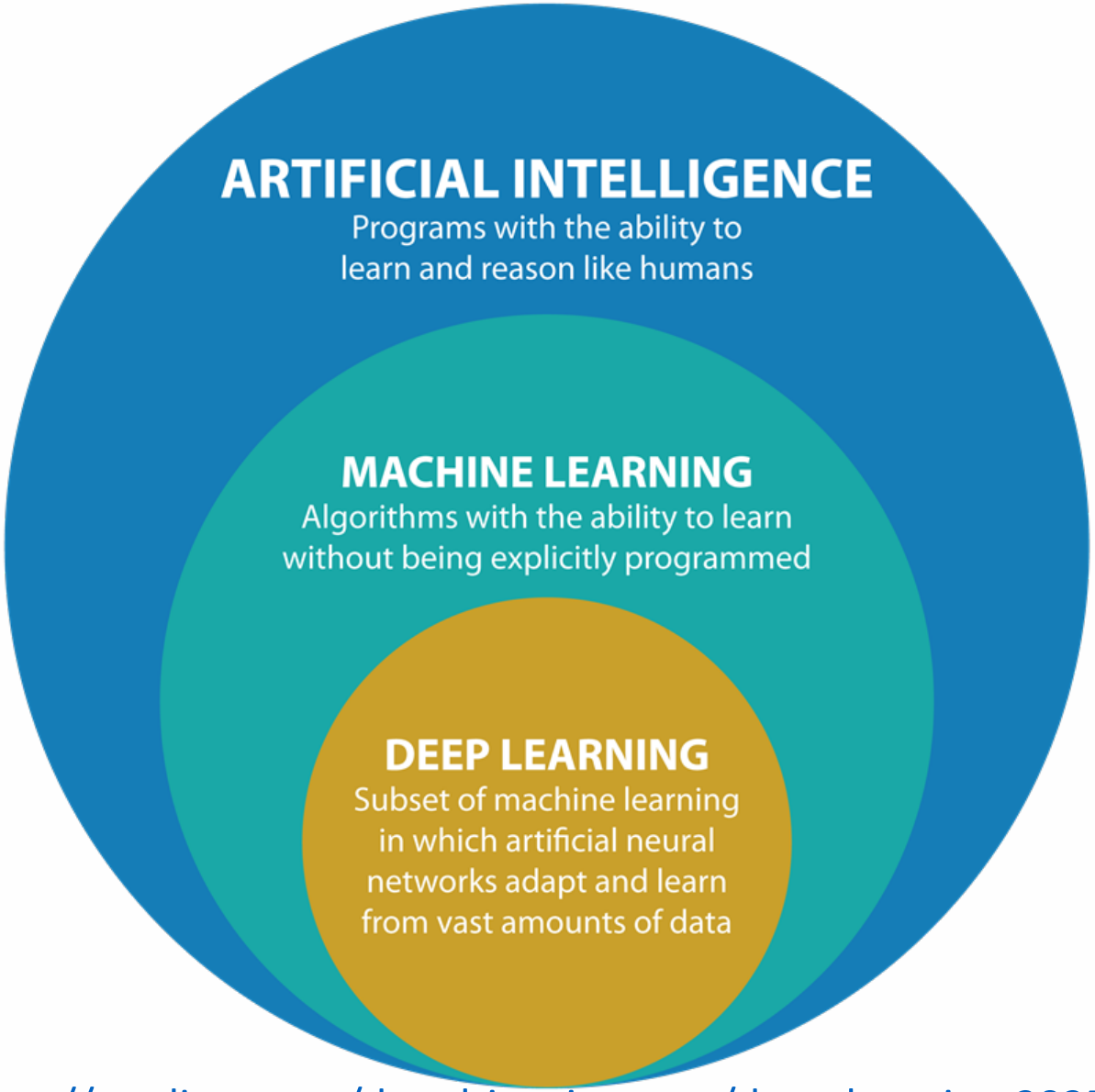
The **training set** is the subset of data that is used to train the ML model(s)

The **validation set** is generally used during Model selection and hyperparameter tuning

Hyperparameters are those parameters that are set before training a model

The **test set** is used at the end to evaluate the generalization error of the chosen model





https://medium.com/datadriveninvestor/deep-learning-2025e8c4a50

Let's get set up

We won't do any programming today, but we will get you set up with Slack and Jupyter so we can dive into Python next week.

- **Anaconda + Jupyter – our programming environment**
- **Git – our source code management system**
- **Slack – our class discussion forum**

Exercise 1

Matrices

A matrix is a 2-dimensional rectangular array

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

Matrix size = (#rows, #cols) = (3, 2)

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

Matrices can be represented in python as 2-dimensional NumPy arrays

```
import numpy as np
mat = np.ones(2, 3)
```

Matrix operations

Addition: $C = A + B \Rightarrow c_{ij} = a_{ij} + b_{ij}$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \\ a_{31} + b_{31} & a_{32} + b_{32} \end{bmatrix}$$

Element-wise (Hadamard) multiplication:

$C = A \odot B \Rightarrow c_{ij} = a_{ij} \times b_{ij}$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11} \times b_{11} & a_{12} \times b_{12} \\ a_{21} \times b_{21} & a_{22} \times b_{22} \\ a_{31} \times b_{31} & a_{32} \times b_{32} \end{bmatrix}$$

Matrix multiplication

- Prerequisite: A is N × P and B is P × M. C will be N × M
  - The number of rows of A must equal the number of columns of B

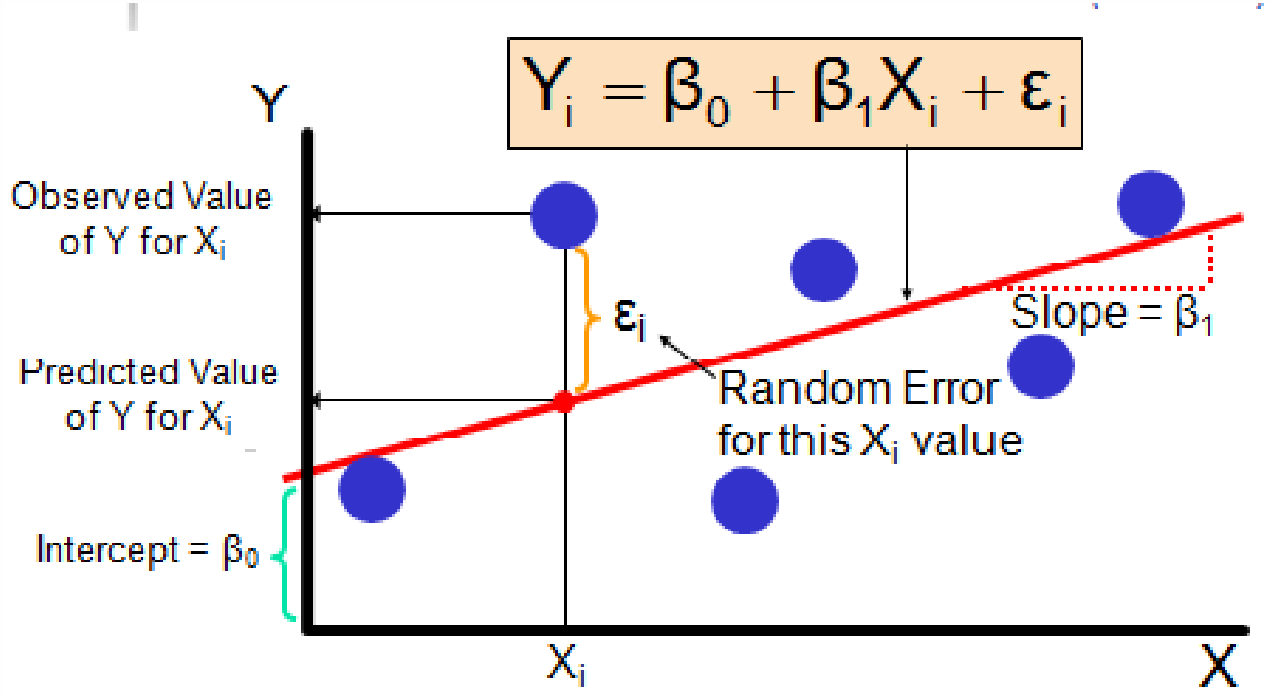
$$C = A \times B \Rightarrow c_{ij} = \sum_{k=1}^{P} a_{ik} \times b_{kj}$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{bmatrix}$$

Linear Regression

- Find the best linear model that fits our data
- This means finding two parameters: slope (β1) and intercept (β0)



Once trained, we can use the model to make predictions => machine learning!!

Linear Regression: Ordinary Least Squares

$$y_1 = \beta_0 + \beta_1 x_1 + \epsilon_1$$
$$y_2 = \beta_0 + \beta_1 x_2 + \epsilon_2$$
$$y_3 = \beta_0 + \beta_1 x_3 + \epsilon_3$$
$$\dots$$
$$y_n = \beta_0 + \beta_1 x_n + \epsilon_n$$

Converted to matricial form:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

where:

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ & \dots \\ 1 & x_n \end{pmatrix} \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{pmatrix} \quad \boldsymbol{\beta} = (\beta_0 \quad \beta_1) \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \dots \\ \epsilon_n \end{pmatrix}$$

Ordinary Least Squares Solution

$$\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{\epsilon}$$

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X\theta}\|^2$$

Find the value of $\boldsymbol{\theta}$ that minimizes the squared sum of the estimation errors $\boldsymbol{\epsilon}$

$$\underline{\boldsymbol{X}}^{\boldsymbol{T}}\,\underline{\boldsymbol{X}}\,\hat{\boldsymbol{\theta}} = \underline{\boldsymbol{X}}^{\boldsymbol{T}}\,\underline{\boldsymbol{y}}$$

$$\hat{\boldsymbol{\theta}} = \left(\underline{\boldsymbol{X}}^{\boldsymbol{T}}\,\underline{\boldsymbol{X}}\right)^{-1}\underline{\boldsymbol{X}}^{\boldsymbol{T}}\,\underline{\boldsymbol{y}}$$

Weekly evaluation form

Please take 2 minutes at the end of the class to fill out the class evaluation form.

Thanks!

https://forms.gle/mY2DmZzN3STatrE66