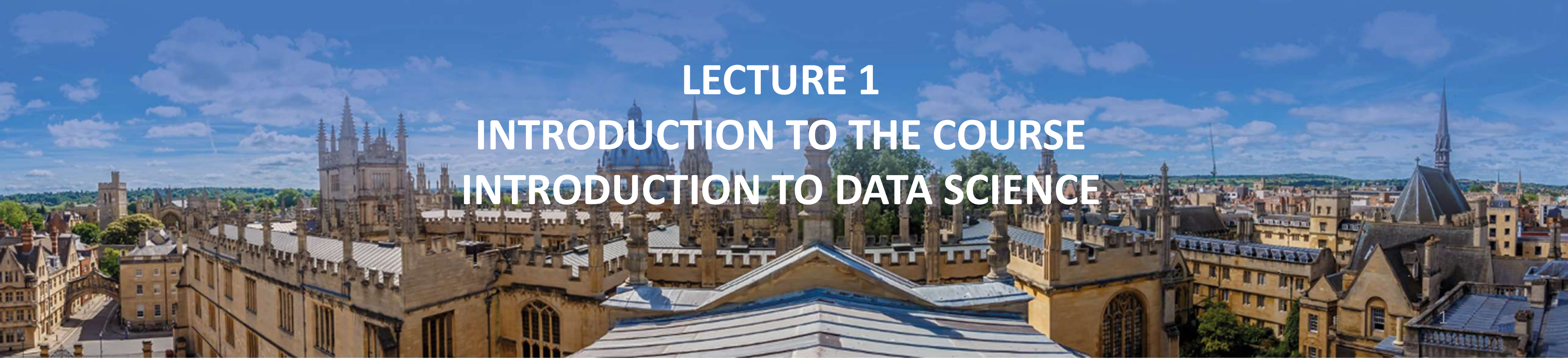
Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

PYTHON PROGRAMMING FOR DATA SCIENCE – Intermediate

Jan – Apr, 2025



LECTURE 1
INTRODUCTION TO THE COURSE
INTRODUCTION TO DATA SCIENCE

Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

Welcome to “Python Programming for Data Science”

- Developed for the Department of Continuing Education

https://github.com/grobles2/p4ds

- Tutors: Nazneen Ali

  Gisela Robles Aguilar

- Email: gisela.aguilar@ndm.ox.ac.uk
- Tue 7-9pm Room 8, Ewert House
- Slack (https://pp4ds-ox-workspace.slack.com/)

Week 1

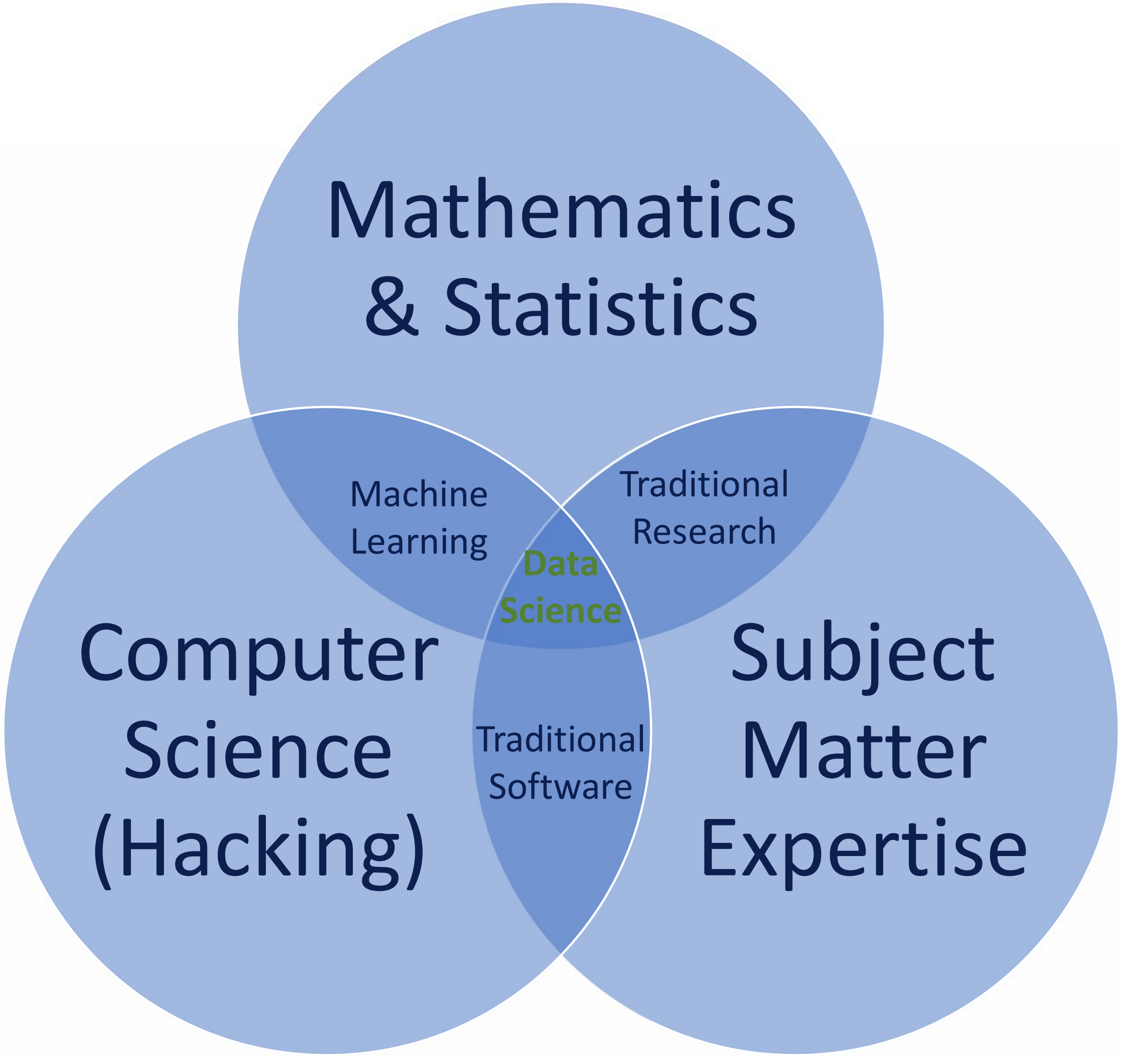
- Introduction to the course
- What is Data Science? (the short of it)
- Introduction to Machine Learning
- Exercise: Matrices and a Linear Regression Example

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

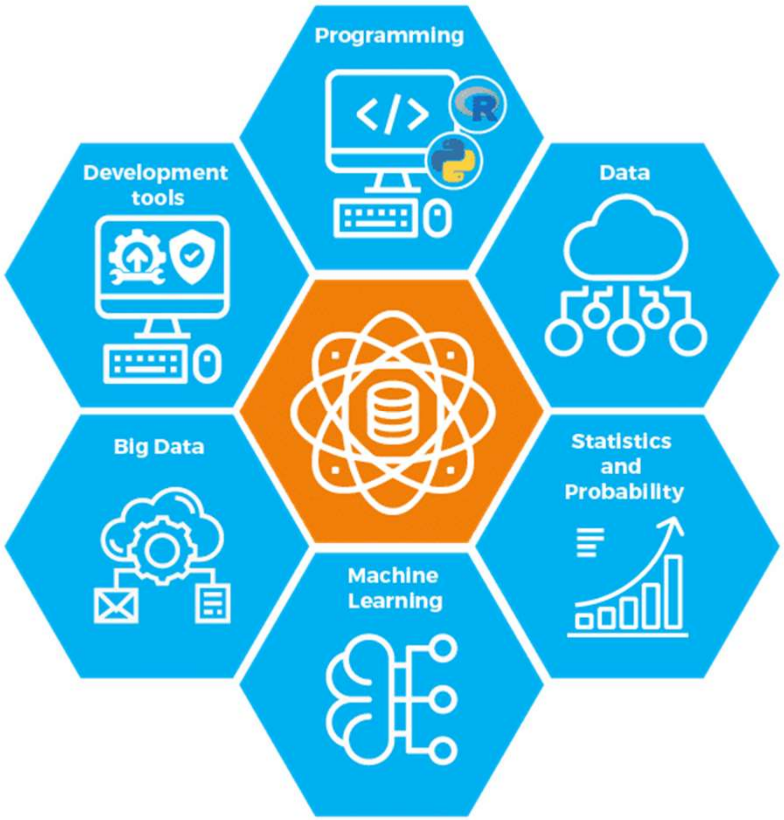
DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD



DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

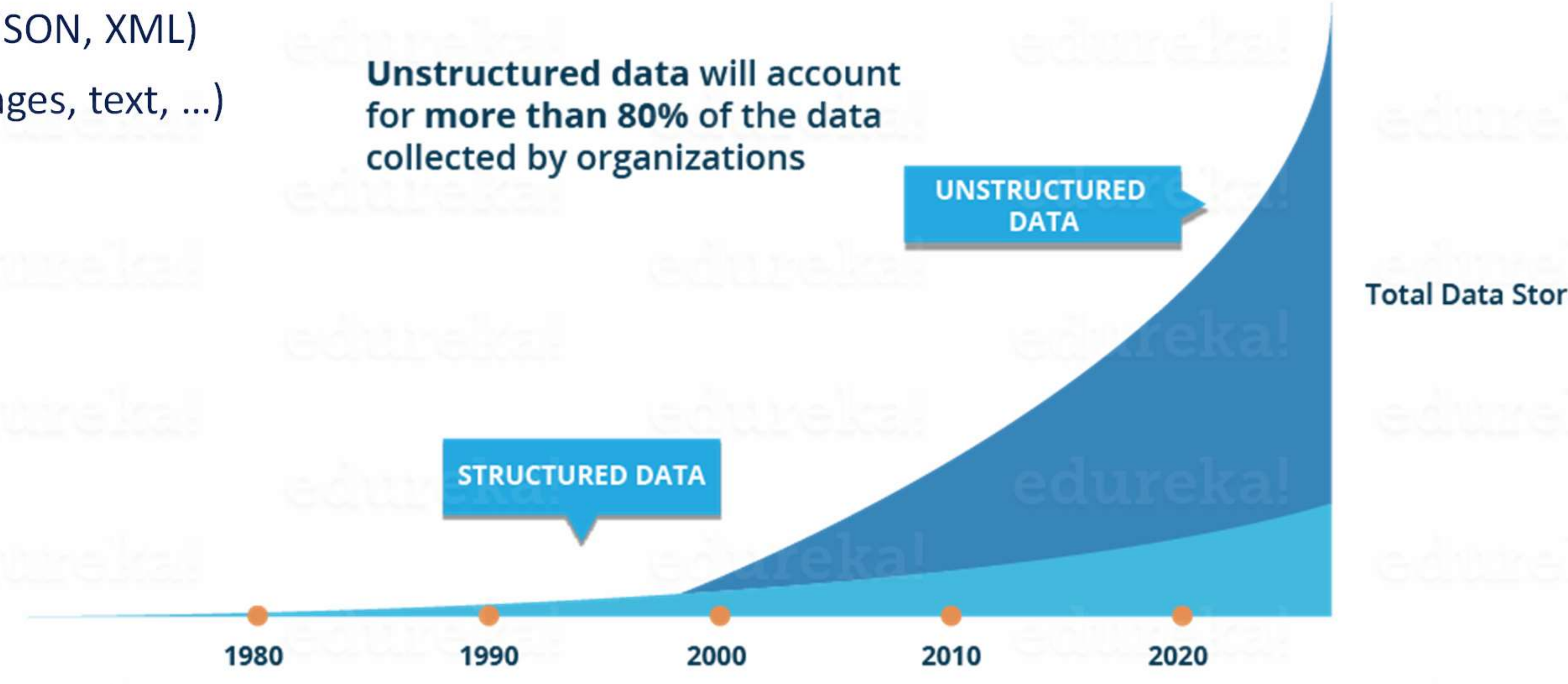
Data Science Life Cycle

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



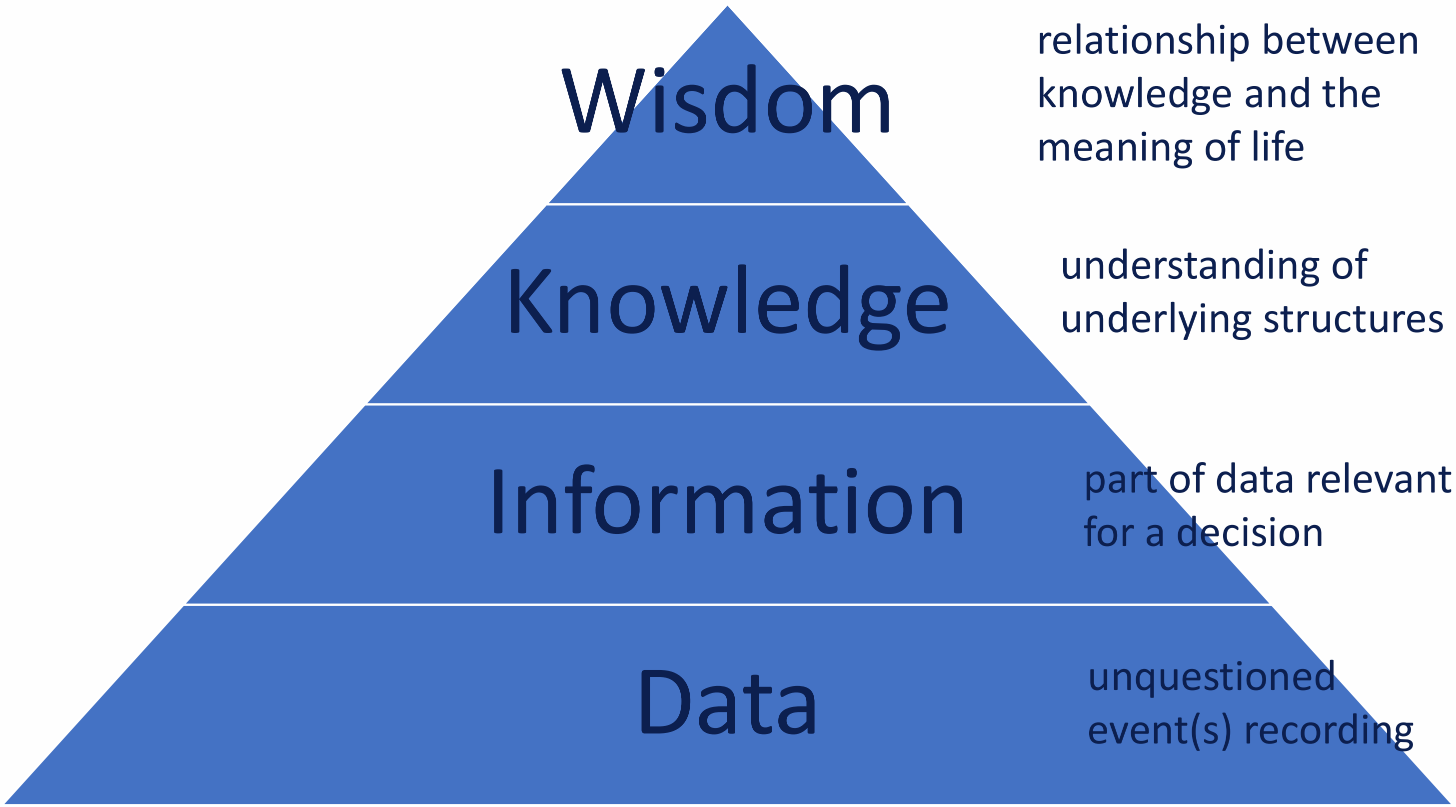
Data

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



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

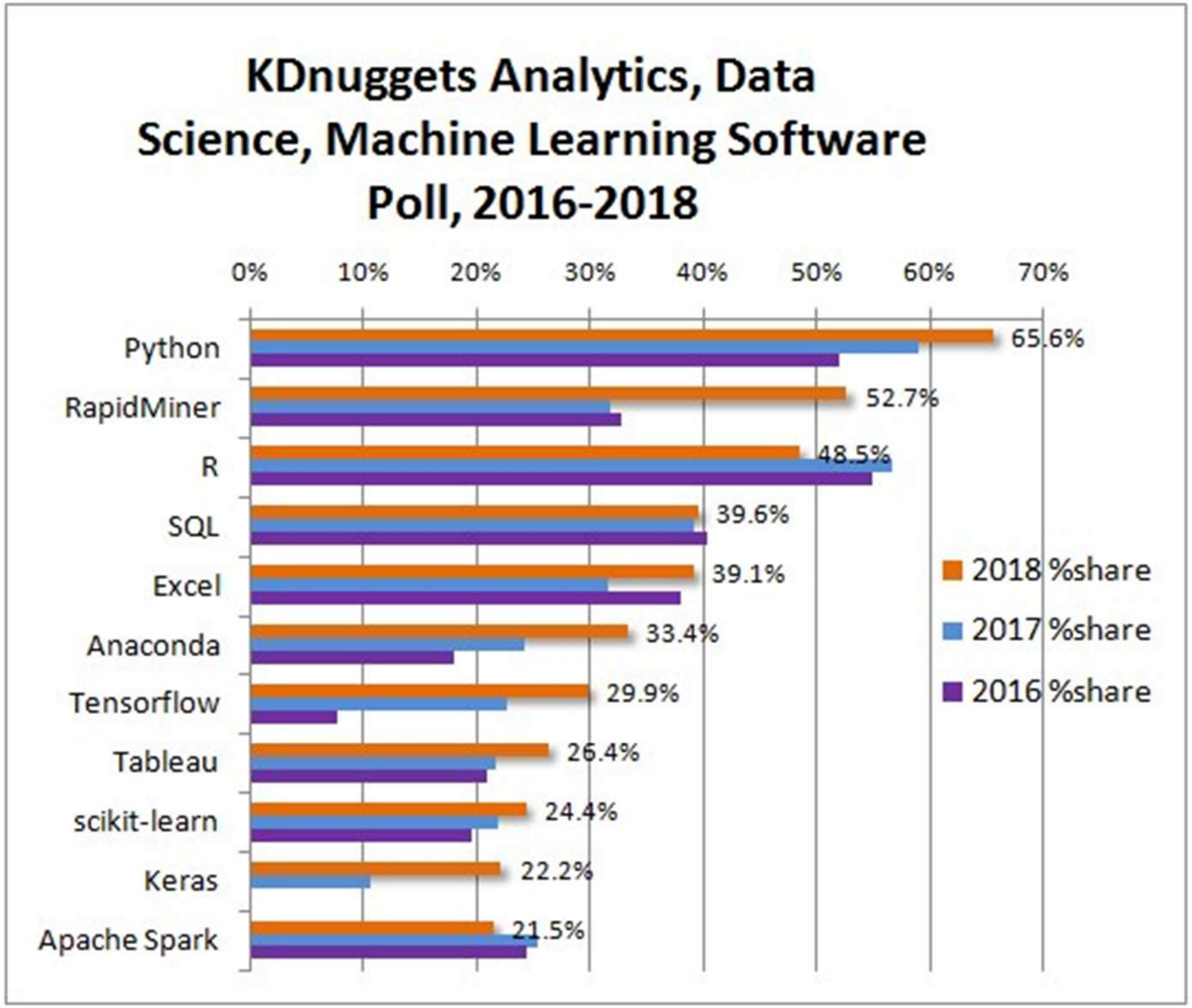
DIKW (epistemological) pyramid



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

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

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

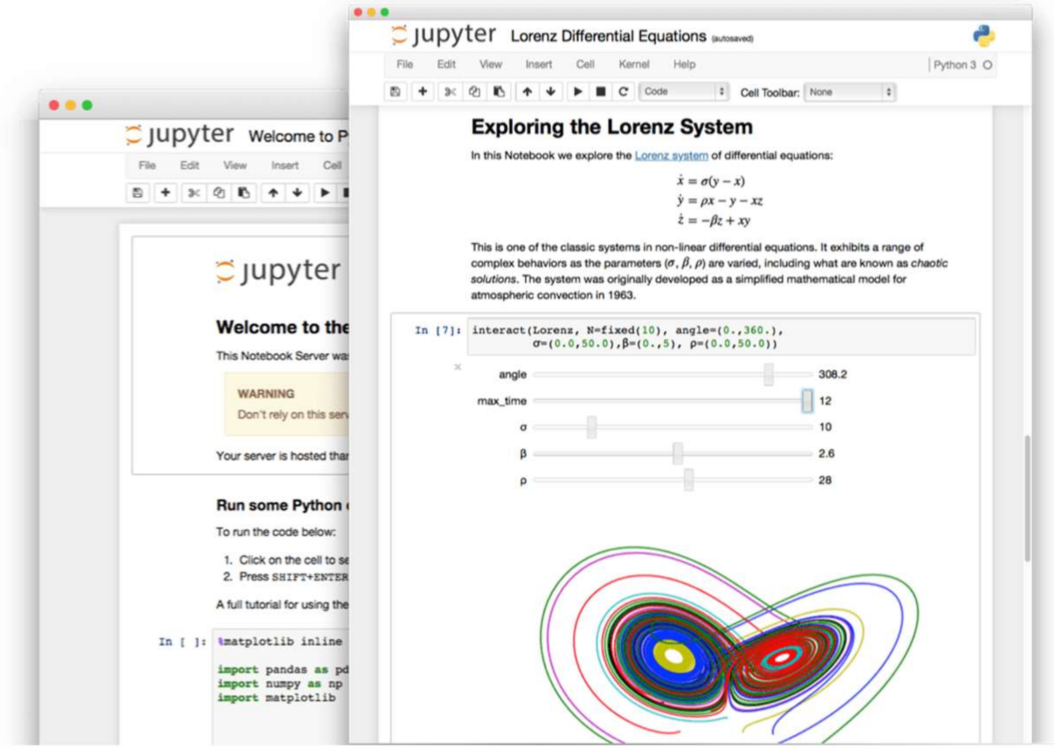
- Lectures
- Practical programming exercises, using Python
- Class discussions, off and online
- A record of attendance is kept
- Summative Assignment: a portfolio of exercises
- Week-to-week class evaluation forms

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/

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

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://github.com/grobles2/p4ds**
- Start on your machine Anaconda Navigator
- Start Jupyter notebook from the interface
- Open the notebook (guidance will be provided)

This week and beyond…

Week 1: Introduction to the course. Basic overview of Machine Learning. Linear Regression example.

Week 2: Overview of a data-science pre-processing pipeline. Exploratory Data Analysis

Week 3: Data cleaning and preparation.

Week 4: Supervised Learning: regression.

Week 5: Supervised Learning: classification.

Week 6: Decision Trees. Ensemble Methods. Hyperparameter Tuning

Week 7: The Perceptron. Back-propagation. Fully-connected neural networks

Week 8: Dimensionality reduction and Unsupervised Learning.

Week 9: Deep Learning: fundamental concepts. Transformers and attention.

Week 10: Deep Learning: other architectures- GANs/Autoencoders

Weekly Oxford Worldwide

DEPARTMENT FOR CONTINUING EDUCATION

UNIVERSITY OF OXFORD

PYTHON PROGRAMMING FOR DATA SCIENCE – Intermediate

LECTURE 1
INTRODUCTION TO MACHINE LEARNING

What is Machine Learning?

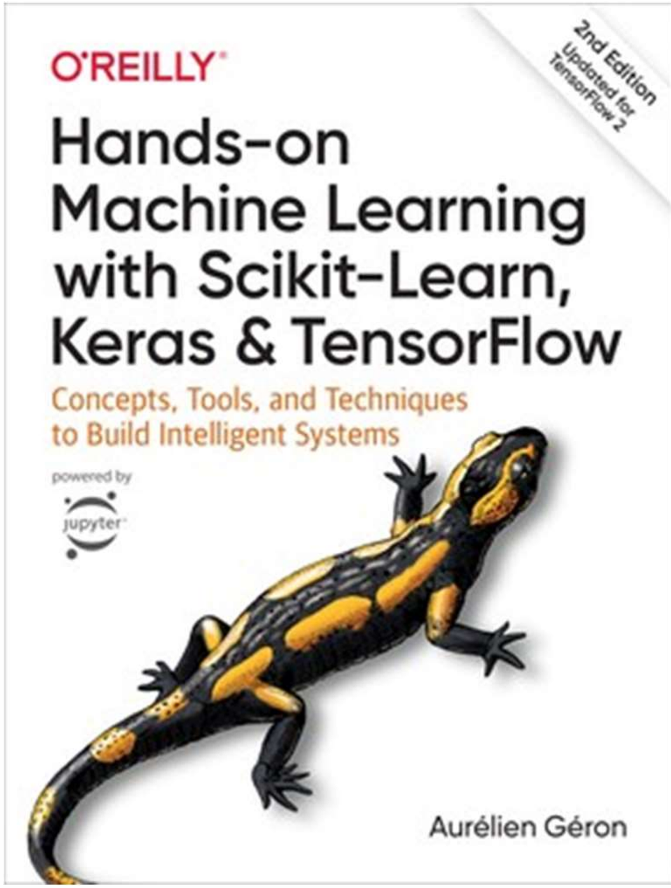
- "[ML is] the field of study that gives computers the ability to learn without being explicitly programmed." (Arthur Samuel, 1959)
- "A computer program is said to learn from experience *E* with respect to some task *T* and some performance measure *P*, if its performance on *T*, as measured by *P*,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

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

“Hands-on Machine Learning” Book

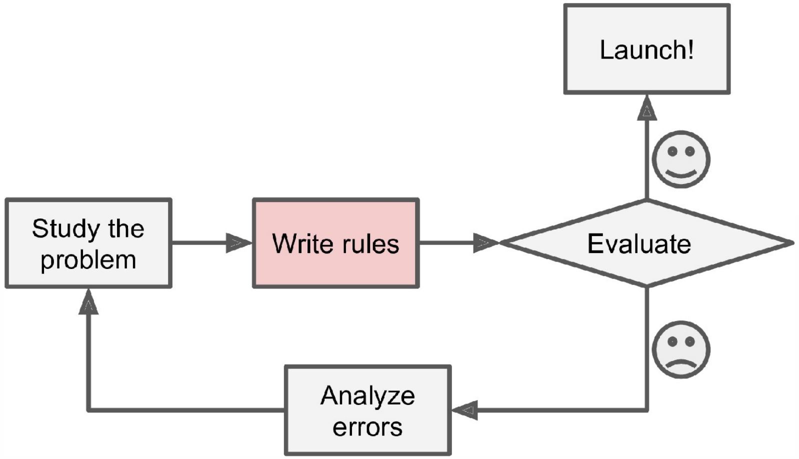
Aurélien Géron***, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems***
O'Reilly Media; 2 edition (October 15, 2019)
ISBN-10: 1492032646
ISBN-13: 978-1492032649

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

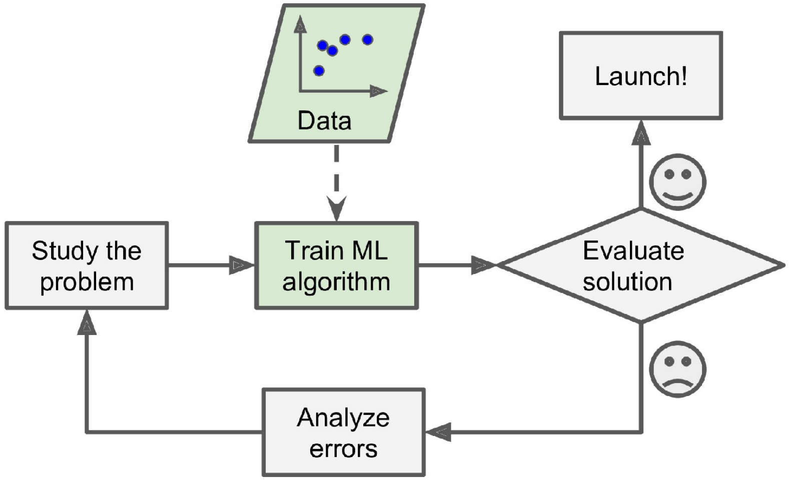


DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

The Traditional Approach



The Machine Learning Approach



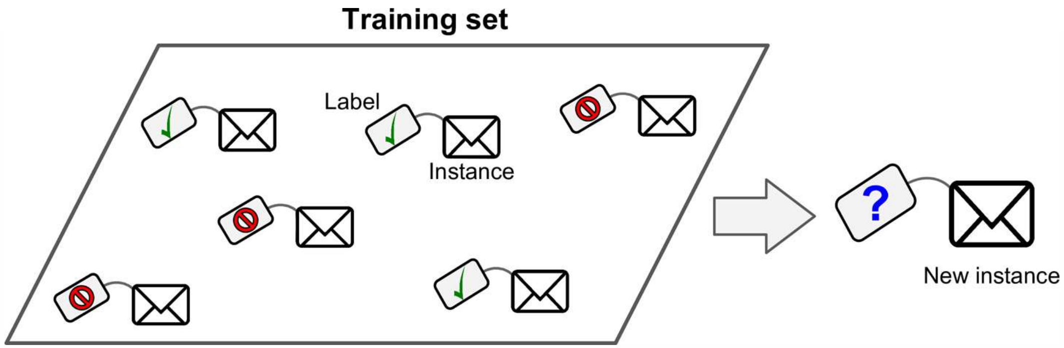
Source: Aurélien Géron, ***Hands-On Machine Learning***

Types of Machine Learning Systems

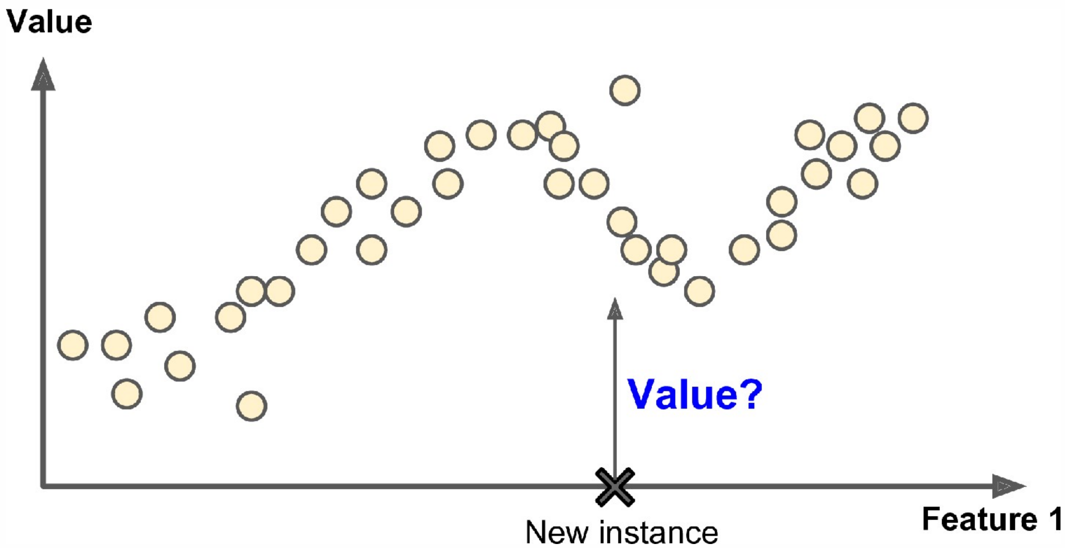
- Are they trained with human supervision? (Supervised, Unsupervised and Reinforcement Learning)
- Can they learn incrementally on the fly? (online versus batch learning)
- Can they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and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

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

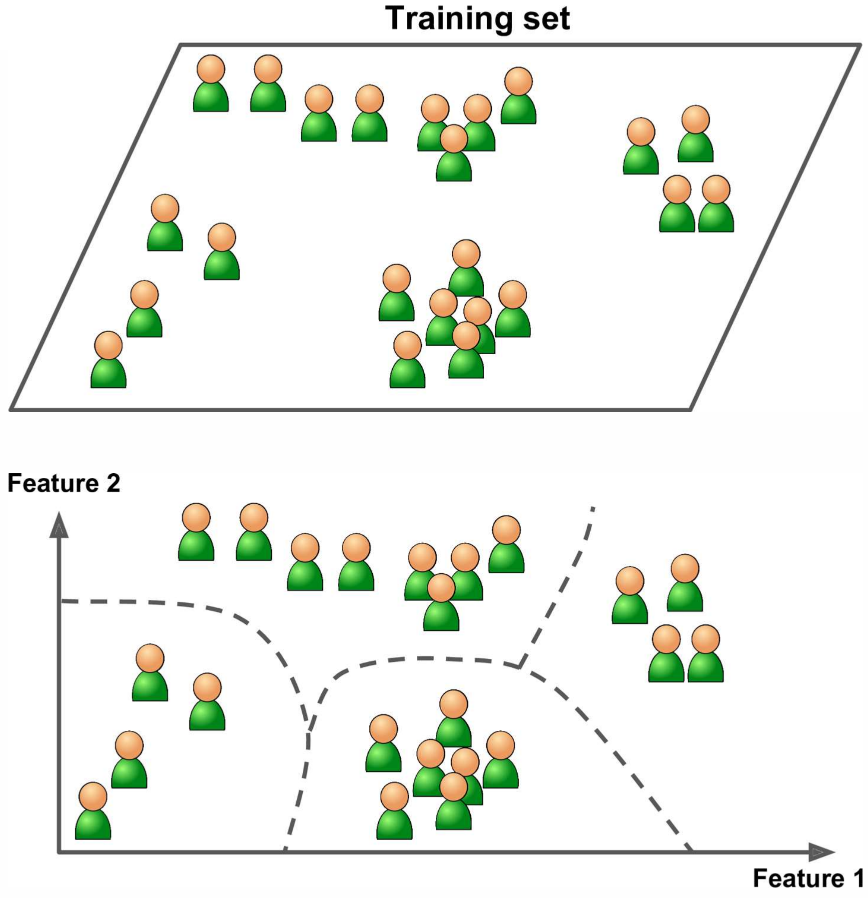
Supervised Learning: examples

- 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

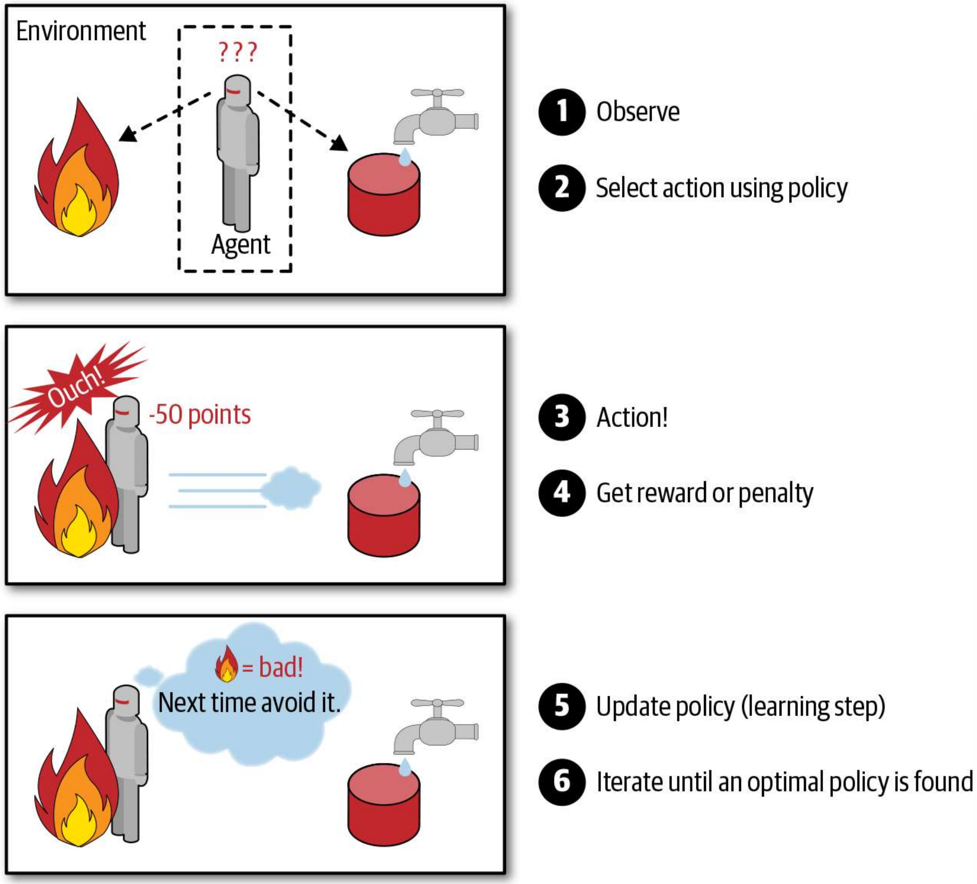
Finding previously unknown patterns in data set without pre-existing labels. Examples:

- 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

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.

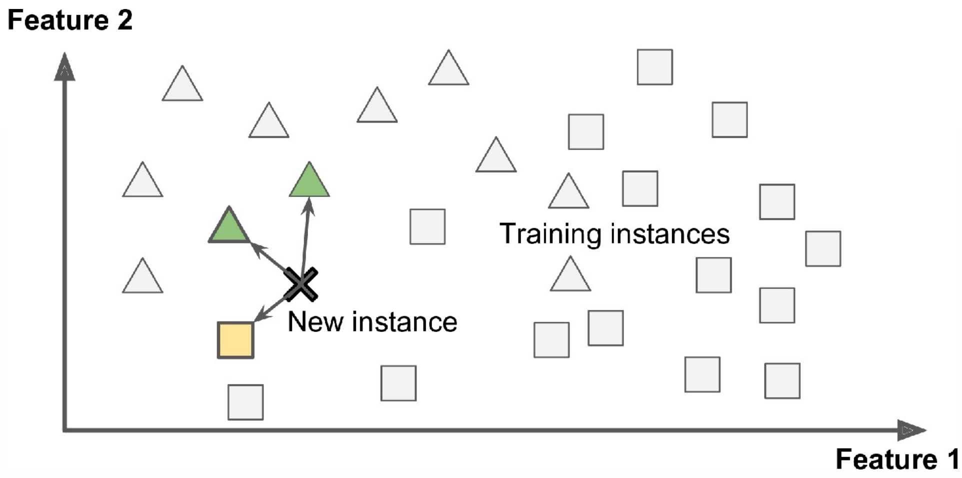
DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

Batch vs Online Learning

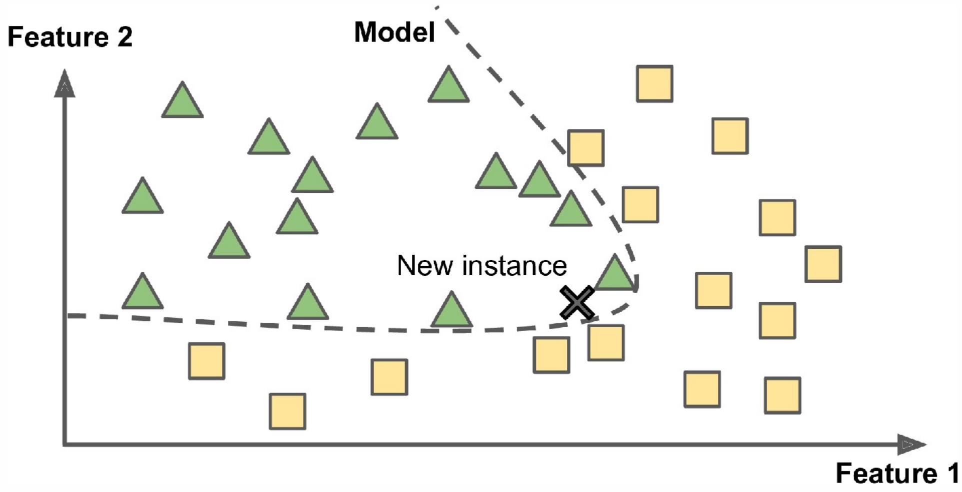
- *batch learning:* system not capable of learning incrementally: it must be trained using all the available data => a lot of time and computing resources => typically done offline. (1) the system is trained, and (2) it is launched into production (no more learning). This is called *offline learning*.
- *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

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

Learn by Instance



Learn by model



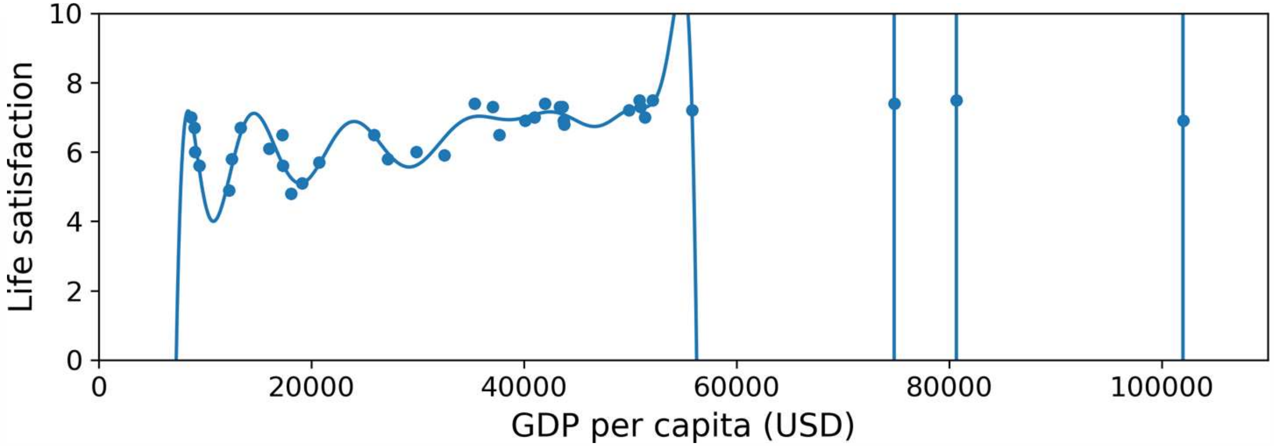
Source: Aurélien Géron, ***Hands-On Machine Learning***

The Unreasonable Effectiveness of Data

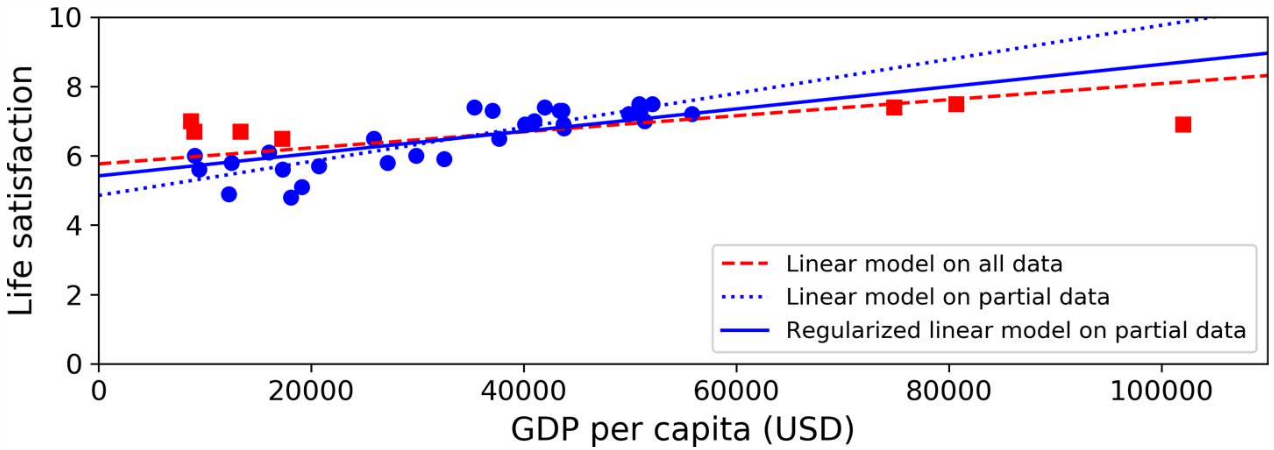
- Different Machine Learning algorithms, including fairly simple ones, performed almost identically on a complex problem if enough data was provided
- “these results suggest that we may want to reconsider the trade-off between spending time and money on algorithm development versus spending it on corpus 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***

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

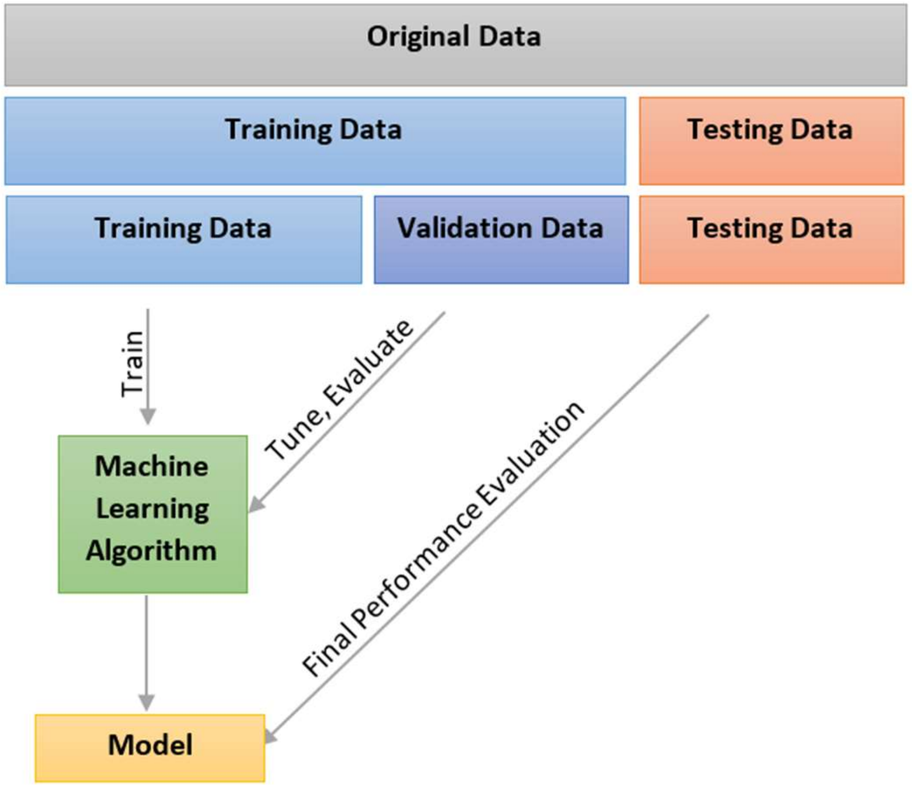
Training, Testing and Validation

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

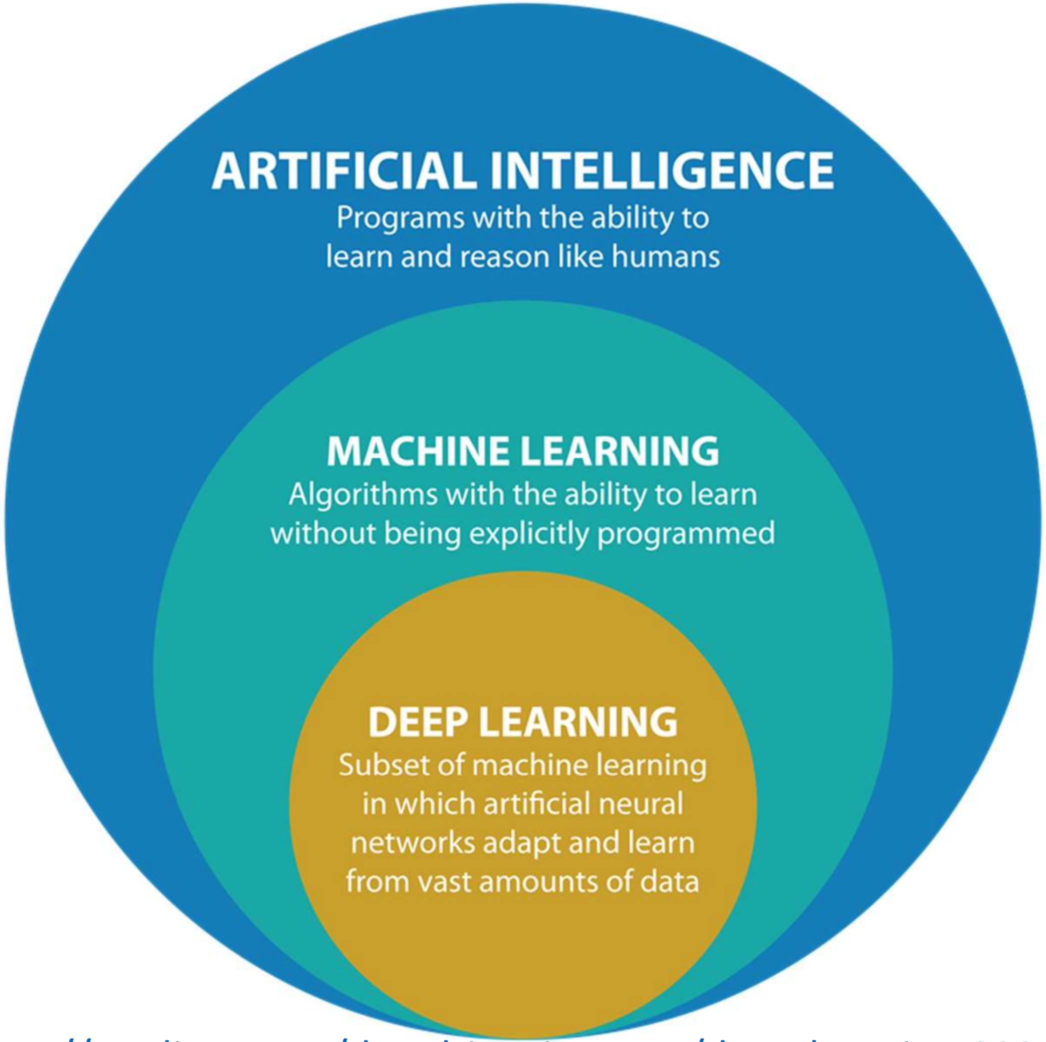
The **validation set** is generally used during Model selection and hyperparameter tuning

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



DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD



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.

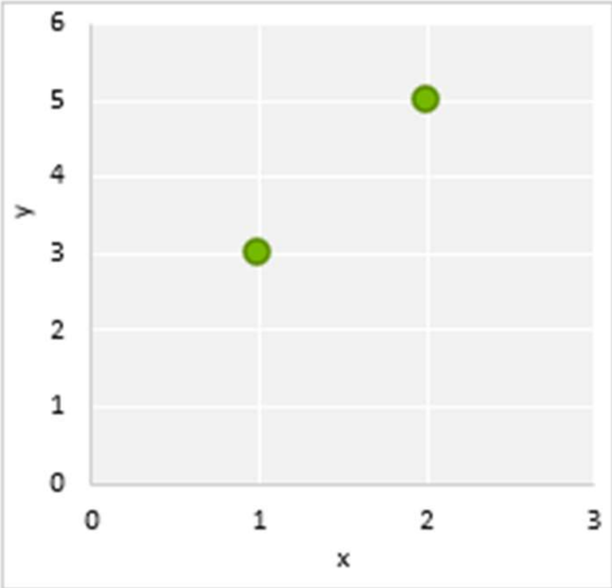
- **Anaconda + Jupyter – our programming environment**
- **Git – our source code management system**
- **Slack – our class discussion forum**

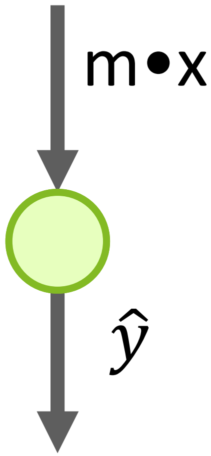
Linear Regression example

A Simpler Model

$$y = mx + b$$

x	y
1	3
2	5





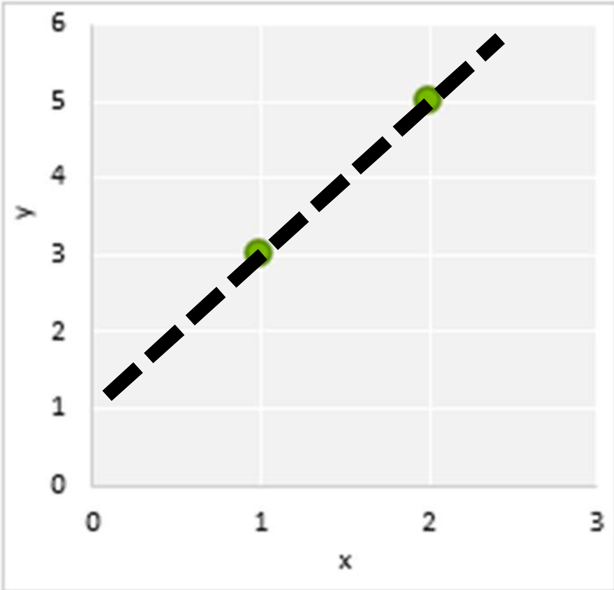
$m = ?$

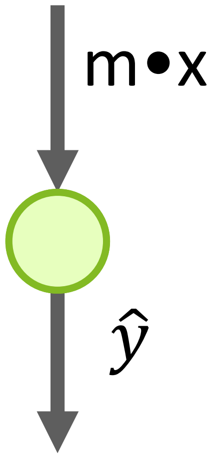
$b = ?$

A Simpler Model

$$y = mx + b$$

x	y
1	3
2	5





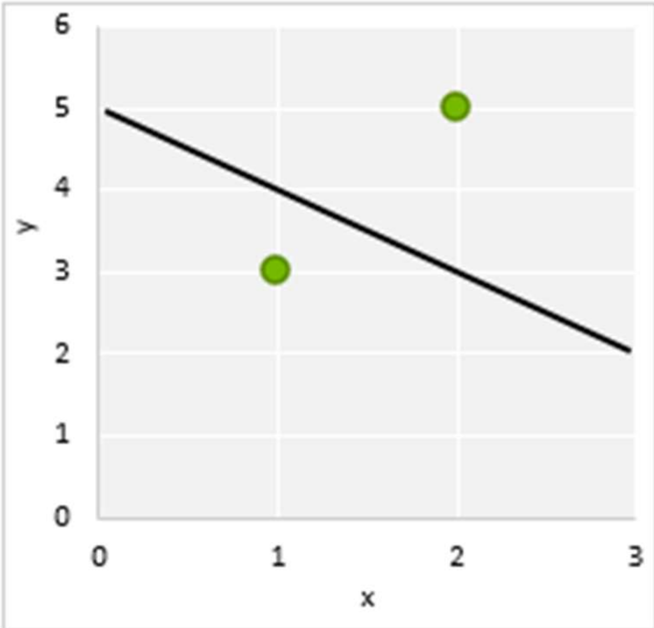
$$m = ?$$

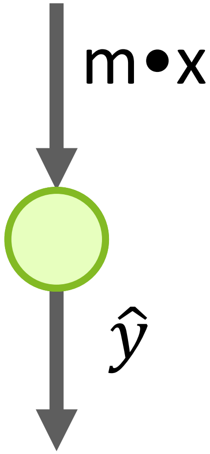
$$b = ?$$

A Simpler Model

$$y = mx + b$$

x	y	$\hat{y}$
1	3	4
2	5	3



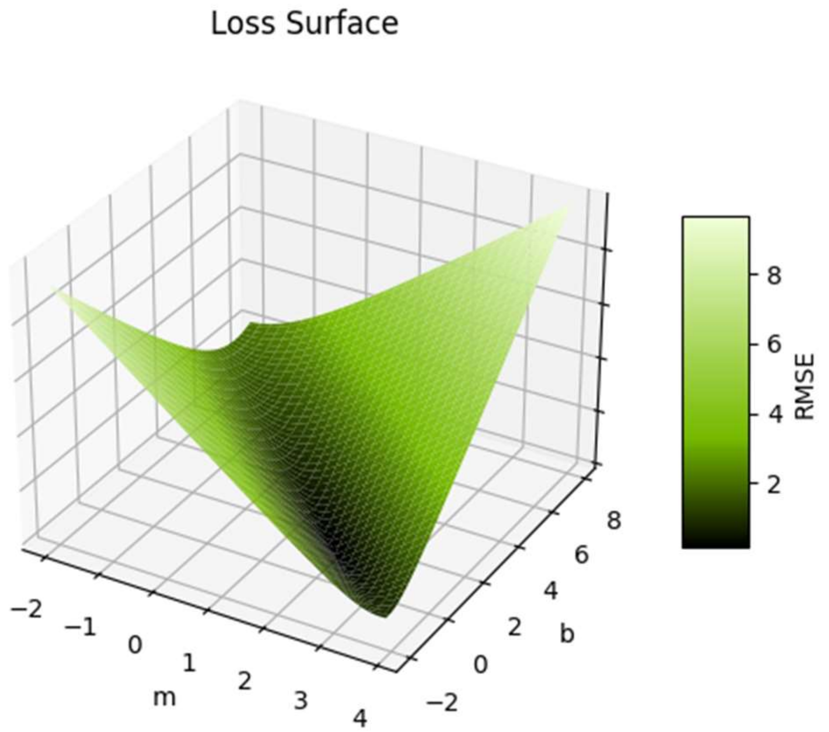


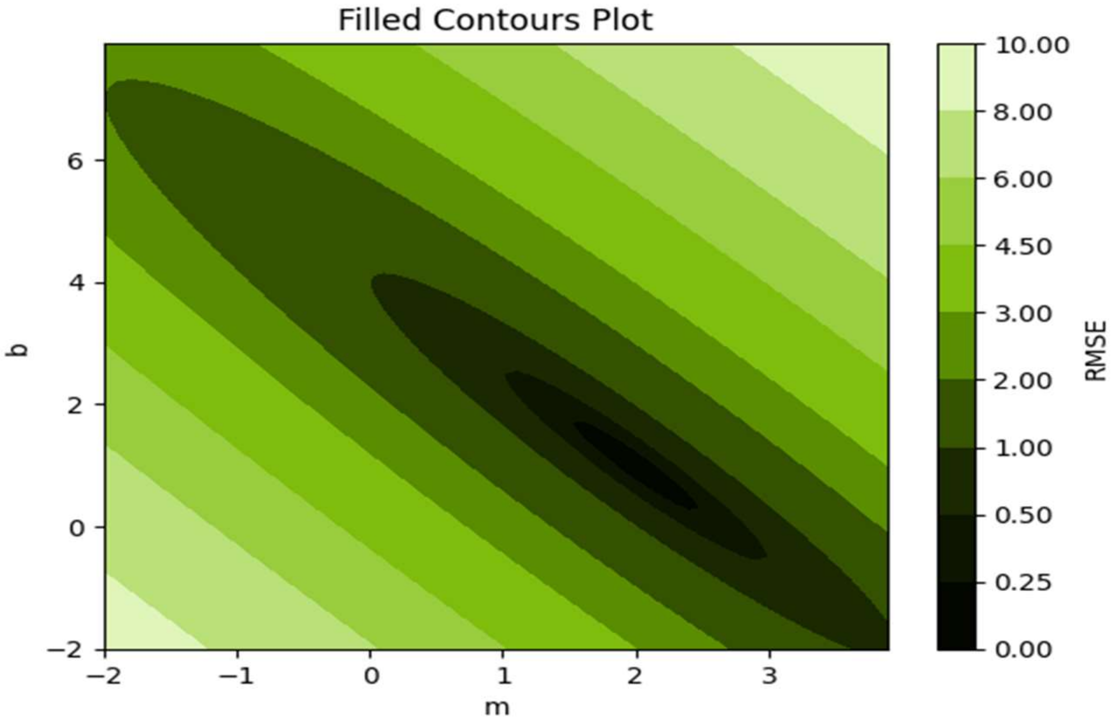
Start Random

$$m = -1$$

$$b = 5$$

The Loss Curve

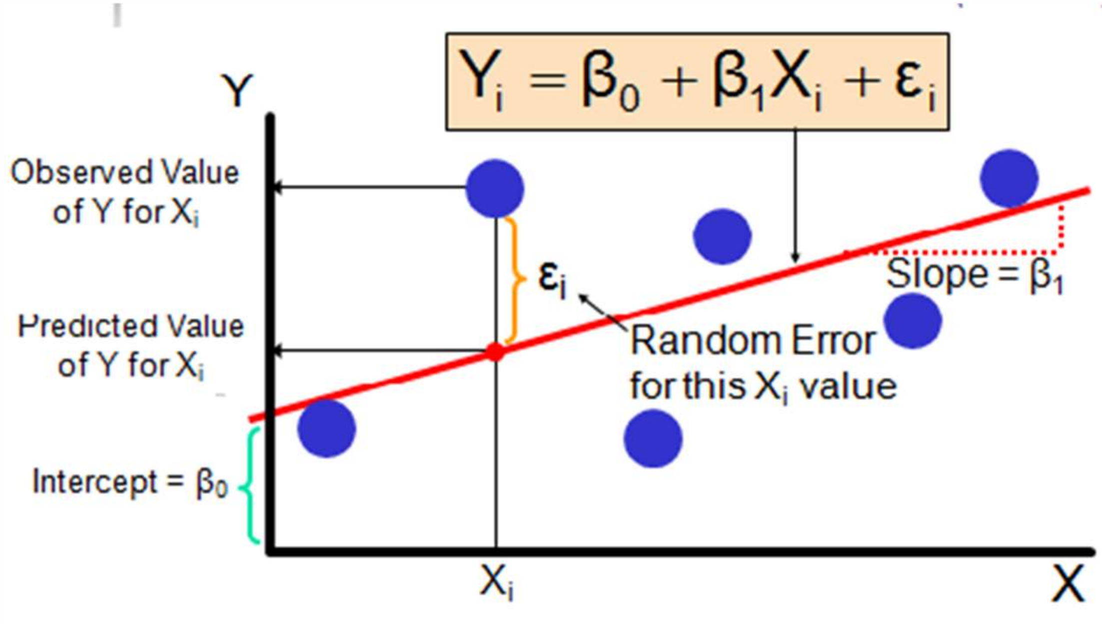




DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

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!!

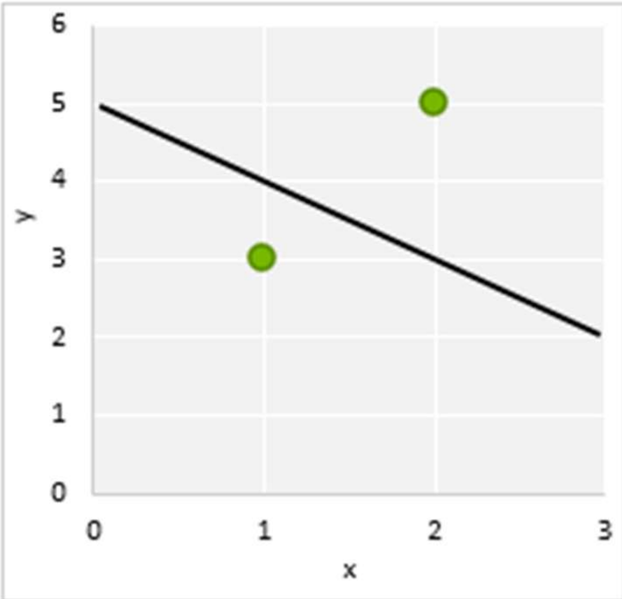
A Simpler Model

$$y = mx + b$$

x	y	$\hat{y}$	err^2
1	3	4	1
2	5	3	4

MSE = 2.5

RMSE = 1.6



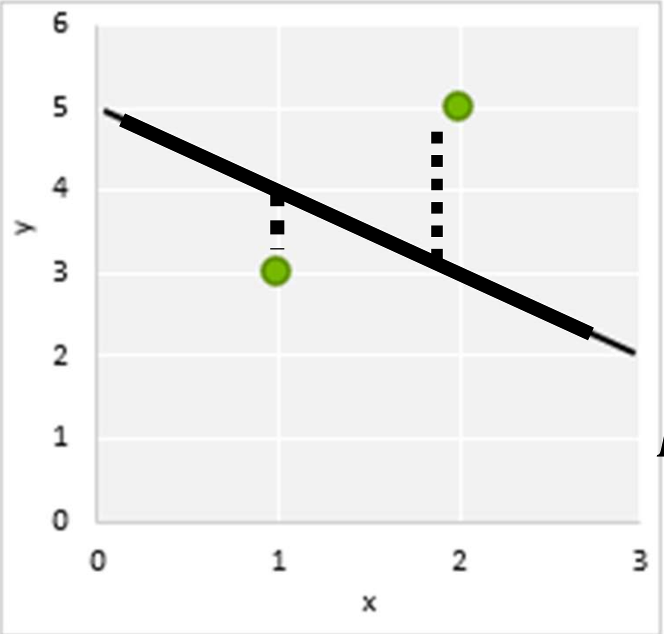

```python
data = [(1, 3), (2, 5)]
m = -1
b = 5


def get_rmse(data, m, b):
    """Calculates Mean Square Error"""
    n = len(data)
    squared_error = 0
    for x, y in data:
        # Find predicted y
        y_hat = m*x+b
        # Square difference between
        # prediction and true value
        squared_error += (
            y - y_hat)**2
    # Get average squared difference
    mse = squared_error / n
    # Square root for original units
    return mse ** .5
```

A Simpler Model

$$y = mx + b$$

x	y	$\hat{y}$	err^2
1	3	4	1
2	5	3	4
		MSE =	2.5
		RMSE =	1.6



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

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:

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

where:

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

DEPARTMENT FOR CONTINUING EDUCATION
UNIVERSITY OF OXFORD

Ordinary Least Squares Solution

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

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{X\theta}\|^2$$

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

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

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

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)

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

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

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

Matrix operations

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

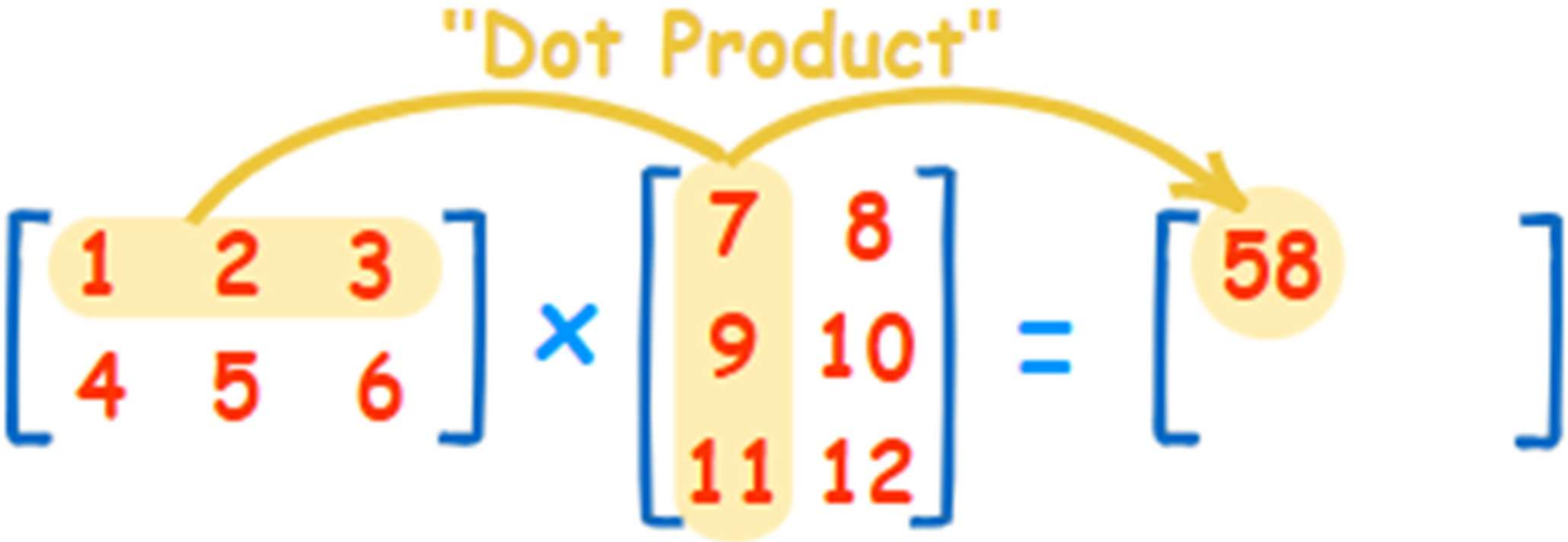
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \\ a_{31} + b_{31} & a_{32} + b_{32} \end{bmatrix}$$

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 $N \times P$ and B is $P \times M$. C will be $N \times M$
  - The number of rows of A must equal the number of columns of B



Weekly evaluation form

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

Thanks!

https://forms.gle/3n94eboCTPB4Avni8