

Machine Learning for Cyber Security

Part 1

Dr Carlos Moreno Garcia

c.moreno-garcia@rgu.ac.uk

Senior Lecturer in Computing | Research Degrees Coordinator

Robert Gordon University, Aberdeen, Scotland, UK

1. Introduction

Objectives

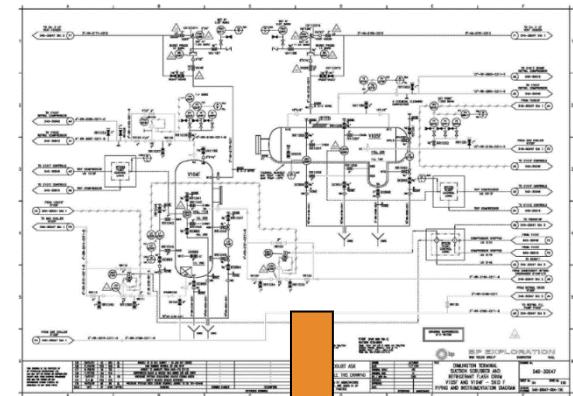
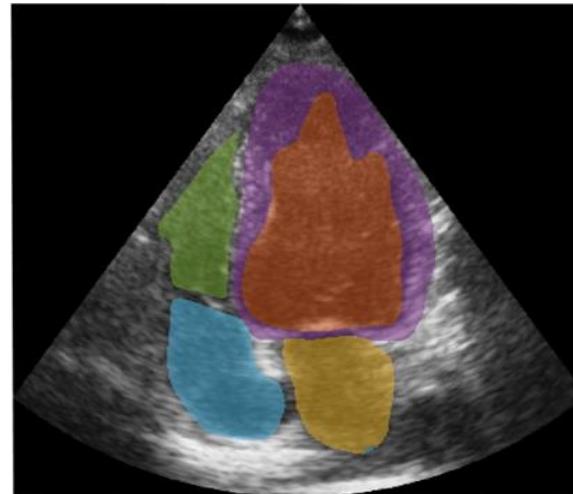
- Understand the main principles of Machine Learning (ML)
- Be exposed to the most relevant tools used in academia and the industry to implement ML-based solutions
- Comprehend a series of real cases where ML is used to solve cybersecurity problems

- 2005-2010: Bachelor of Electronic Engineering (Major in Telecommunications)
 - Tec de Monterrey, Mexico City
- 2010-2012: Teacher/Researcher
 - Research group: Centro de Investigación en Mecánica y Biodiseño (Prof. Rogelio Bustamante)
 - Tec de Monterrey, Mexico City
- 2012-2013: Ms C. Cyber Security and Intelligent Systems
 - Universitat Rovira i Virgili, Tarragona, Spain
- 2013-2016: Ph. D. Computer Science and Mathematics of Security
 - Research group: Sistemas Sensoriales Aplicados a la Industria (Prof. Francesc Serratosa)
 - Universitat Rovira i Virgili, Tarragona, Spain
- 2017-2018: Research Fellow in Computing
 - Research w/ Dr Eyad Elyan and Prof Chrisina Jayne (Industrial collaboration with DNV GL)
 - Robert Gordon University, Aberdeen, UK
- 2018: Lecturer in Computing
 - Placements and Electives Coordinator
 - Robert Gordon University, Aberdeen, UK
- 2020: Senior Lecturer in Computing
 - Research Degrees Coordinator
 - Robert Gordon University, Aberdeen, UK



Main Achievements

- Published 57 scientific papers in international journals and conferences
- Two patents and several prototypes
- Worked with multiple academic partners:
 - UNAM
 - IPN
 - Tec de Monterrey
 - IMP
 - University of Utah
 - University of Munster
 - Taylors's University (Malaysia)
- Collaboration with the industry:
 - Energy sector: DNV GL, Equinor (Statoil), Total, Baker, AISUS, Mintra Group, etc.
 - Healthcare: NHS, ISSSTE
- EuDIF (RedGlobalMX and Google)



Event	Equipment Category	Size	Number
JDY/CELLAR/RJAS/W	Piping	16	
JDY/CELLAR/RJAS/W	Act. Valve	16	0.5
JDY/CELLAR/JASIN/W	Piping	16	
JDY/CELLAR/JASIN/W	Act. Valve	16	0.5
JDY/PROC/JASIN/W	Piping	16	
JDY/PROC/JASIN/W	Act. Valve	16	2
JDY/PROC/JASIN/W	Flange	16	7
JDY/PROC/JASIN/W	Piping	6	
JDY/PROC/JASIN/W	Man Valve	16	3
JDY/PROC/JASIN/W	Piping	2	
JDY/PROC/JASIN/W	Flange	2	2
JDY/PROC/JASIN/W	Inst. Con.	2	2
JDY/PROC/JASIN/W	Man Valve	6	0.5

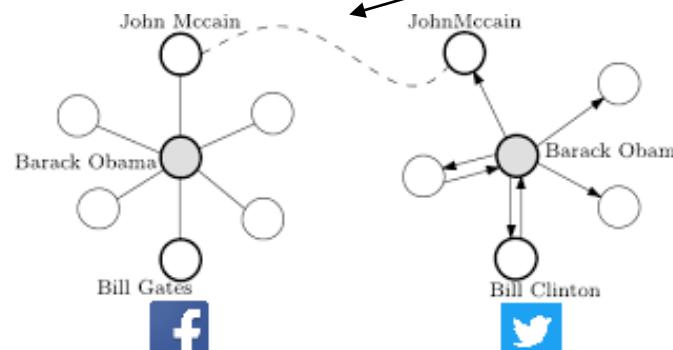


Where am I!?



How did I get “here”?

MSc Cyber Security



Networks



Biometrics

2. Fundamentals of ML

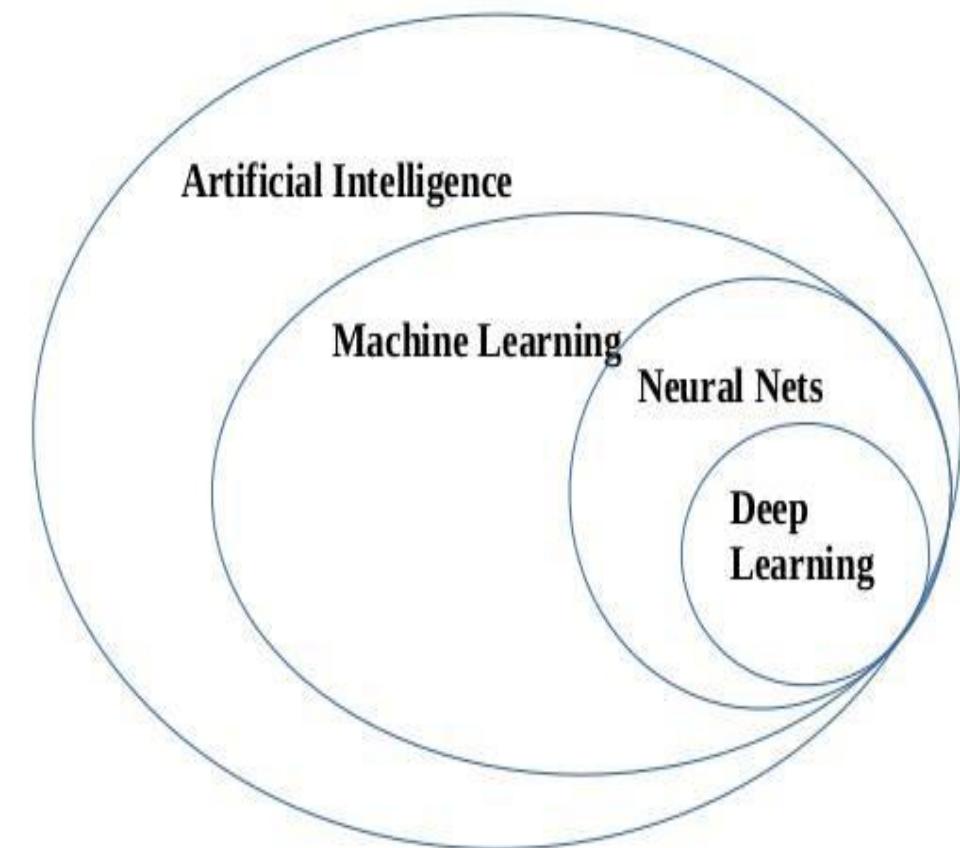
Go to www.menti.com and use the code you see on the screen

What is ML?

- “Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.” — Wikipedia
- “Machine learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world.” — Nvidia
- “Machine learning is the science of getting computers to act without being explicitly programmed.” — Stanford
- **Our working definition**, it’s the statistical discrimination, where the “machine” itself learns the discriminator (mathematical models) using the provided data

What is ML?

- Artificial Intelligence: Whole knowledge field
- Machine Learning: An important part of AI, but **not** the only one
- Neural Networks: One of popular machine learning types
- Deep Learning: A modern way of building, training and using NN. A new architecture



Why ML?

- Imagine you need to buy a house
 - Searched all over, newly built (£400k), year-old (£380k), 2-year-old (£360), 3-year-old (£340), and so on.
 - Price drops by 20k every year, but no less than 100k!
 - Predict the price based on known historical data (regression)
 - But it's not that simple! Different dates, # bedrooms, present condition, location, seasonal demand spikes, etc
 - How many hidden factors are there to determine the house price?
 - An average human can't find these patterns!
- Machine copes with this task much better than a human

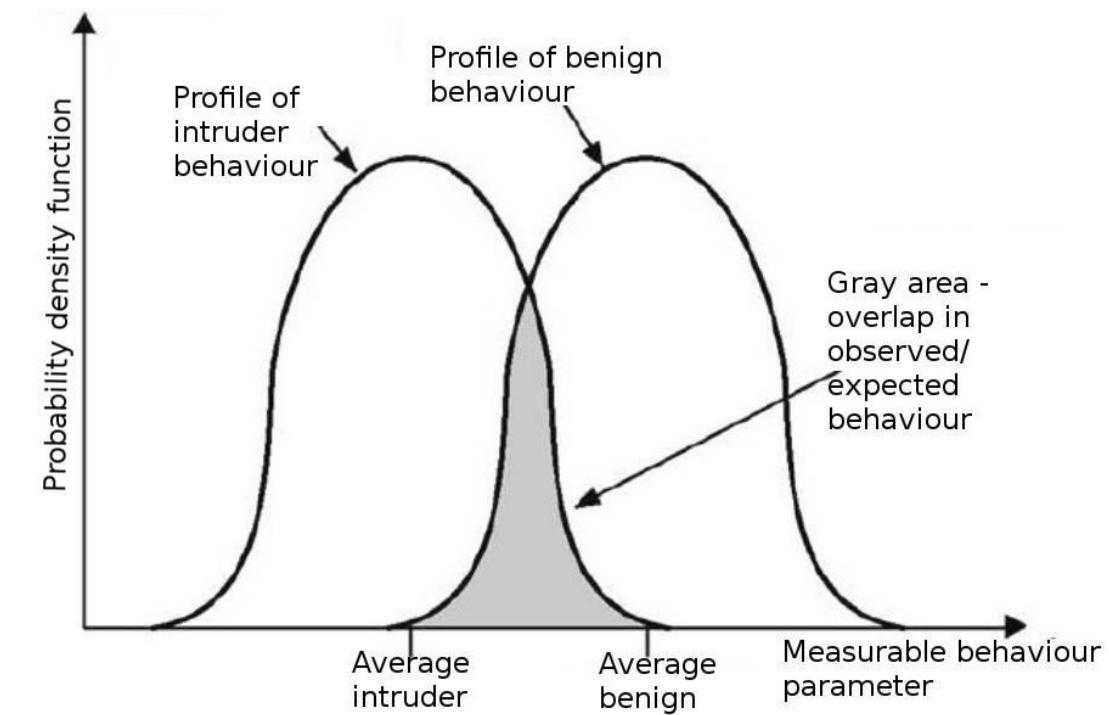


TARGET

<https://www.forbes.com/sites/kashmirihill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/>

How ML fits in Cyber Security?

- Monitoring (user/systems) activities to search for known attacks and/or suspicious activities
- Signature/misuse-based detection
 - Contain a database of recognised attacks
 - Activity is compared with signature database
 - Zero Day go undetected!



How ML fits in Cyber Security?

- Anomaly/behaviour based
 - Use tools borrowed from Machine Learning (ML)
 - Assumption → behaviour differ
 - Have a notion of normal activity
 - Learnt from previously seen benign activities
 - **High false positive rate!**



ML Components

- Goal: Predict results based on incoming data:
 1. Training data
 - Want forecast stocks? Find the price history
 - Want to detect spam? Find samples of spam/non-spams
 - How to gather?
 - Manual (accurate, expensive) and automatic (cheaper)
 - Google use their customers to label data
 - Collecting a good quality dataset is extremely difficult
 - Garbage in, garbage out
 - Companies may be happy to reveal algorithms, but not the data!



ML Components

2. Features

- Individual measurable characteristic, factors for a machine to look at
 - Organised in a table, features are column names
- Choosing right features is a crucial
 - Informative, discriminating and independent
 - Domain knowledge is essential here

3. Algorithms

- Multiple algorithms for one problem (spot-check different algorithms)
- The method you choose affects the precision, performance, and size of the final model
- If the data is bad, even the best algorithm won't help!

Types of Data

- Structured
 - Easily mapped to identifiable column headers
- Unstructured
 - Cannot be mapped
- Semi-structured
 - A mix
- Labelled
 - Contains tags
- Unlabelled
 - Doesn't

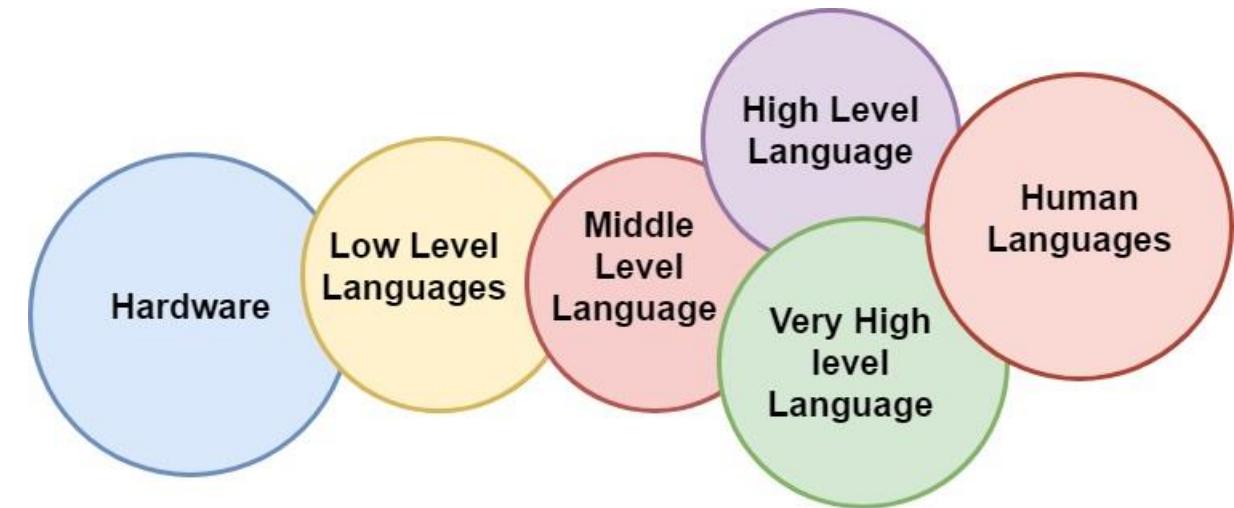
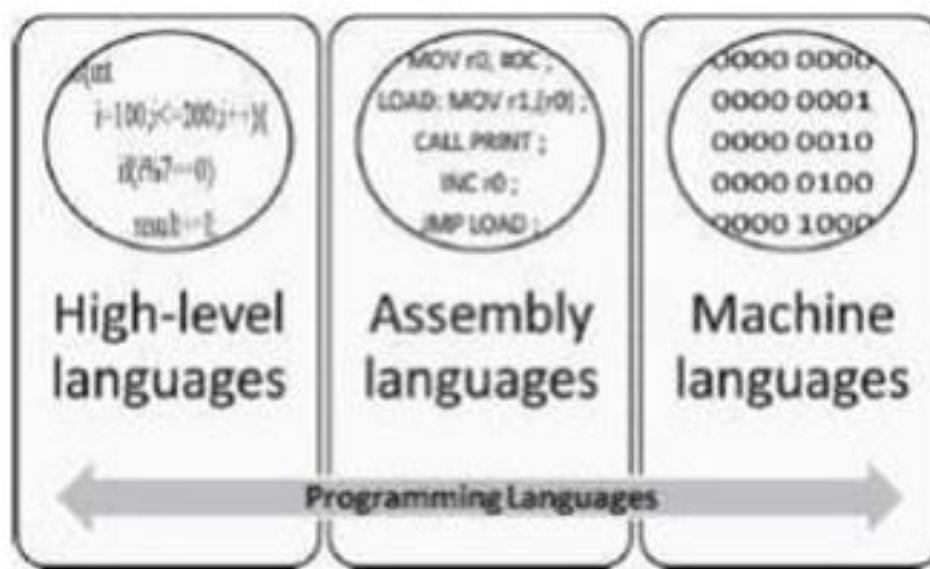
ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit
1000025	5	1	1	1	2	1	3	1	1
1002945	5	4	4	5	7	10	3	2	1
1015425	3	1	1	1	2	2	3	1	1
1016277	6	8	8	1	3	4	3	7	1
1017023	4	1	1	3	2	1	3	1	1
1017122	8	10	10	8	7	10		7	1
1018099	1	1	1	1	2	10	3	1	1
1018561	2	1	2	H	2	1	3	1	1
1033078	2	1	1	1	2	1	1	1	5
1033078	4	2	1	1	2	1	2	1	1

3.1. Tools for ML

Go to www.menti.com and use the code you see on the screen

Types of Programming Languages



Why to Use High-Level?

- Widely used
- Huge growing ecosystems
- Industry is on board!
- Data-driven
- Leads to dashboards
- CONS: Speed
 - May be crucial in CyberSec, but there are ways to solve it



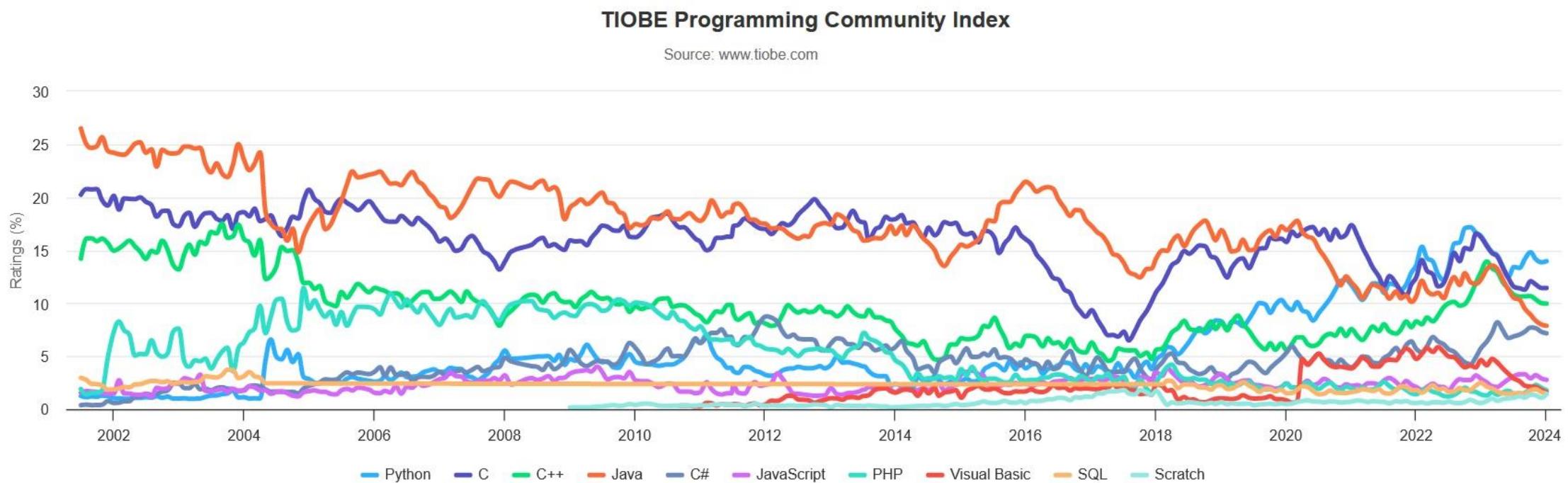
Why ML Top Programming Languages



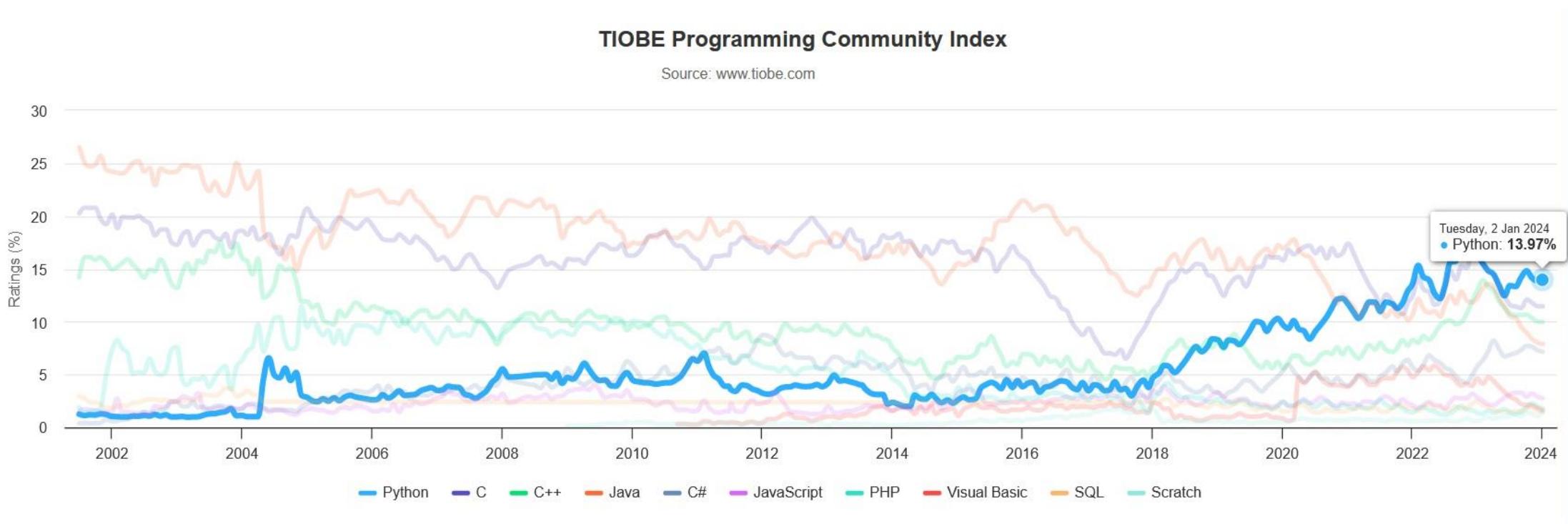
TIOBE Index

May 2024	May 2023	Change	Programming Language	Ratings	Change
1	1		 Python	16.33%	+2.88%
2	2		 C	9.98%	-3.37%
3	4		 C++	9.53%	-2.43%
4	3		 Java	8.69%	-3.53%
5	5		 C#	6.49%	-0.94%
6	7		 JavaScript	3.01%	+0.57%
7	6		 Visual Basic	2.01%	-1.83%
8	12		 Go	1.60%	+0.61%
9	9		 SQL	1.44%	-0.03%
10	19		 Fortran	1.24%	+0.46%

TIOBE Index



TIOBE Index

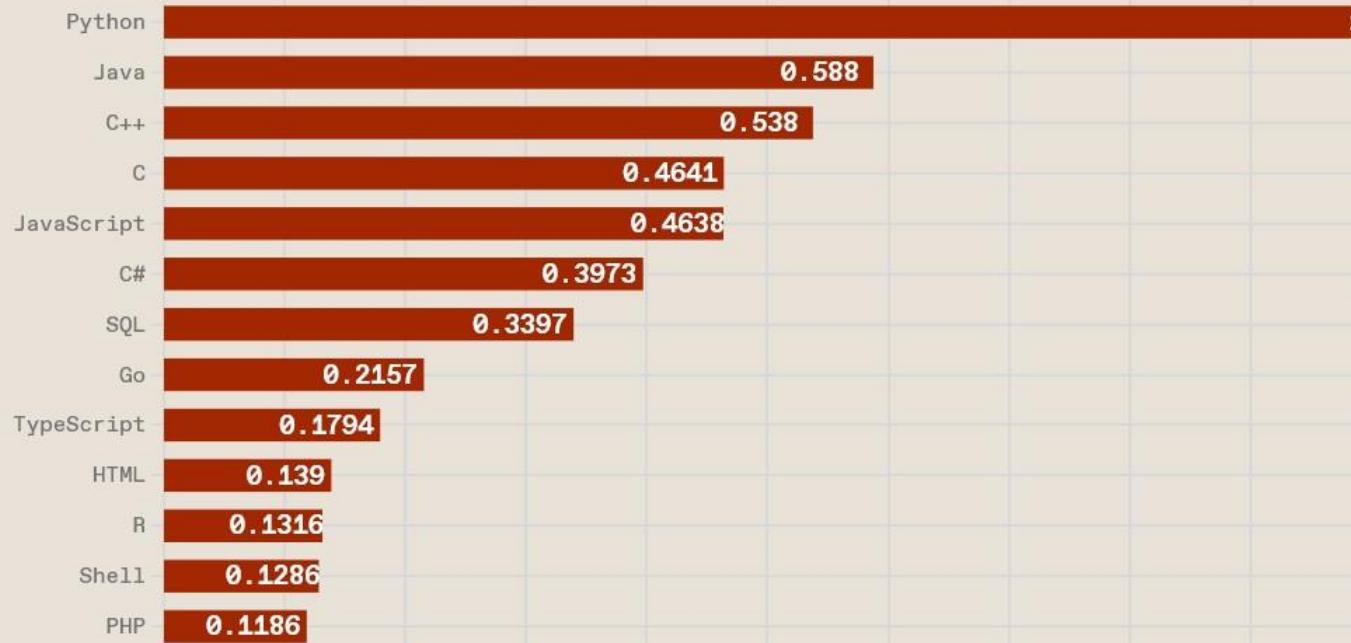


IEEE

Top Programming Languages 2023

Click a button to see a differently weighted ranking

Spectrum Jobs Trending



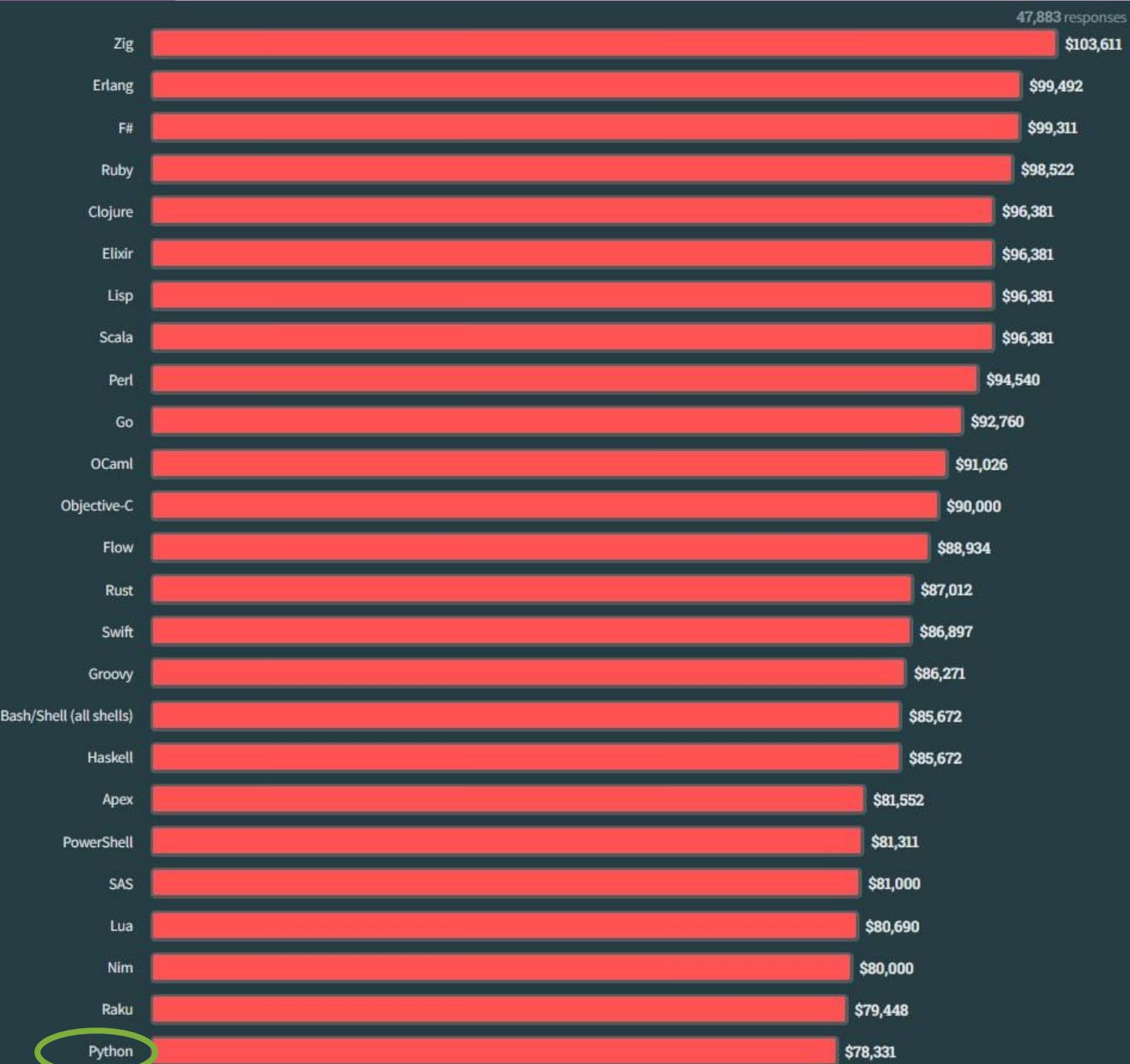
<https://spectrum.ieee.org/top-programming-languages/>

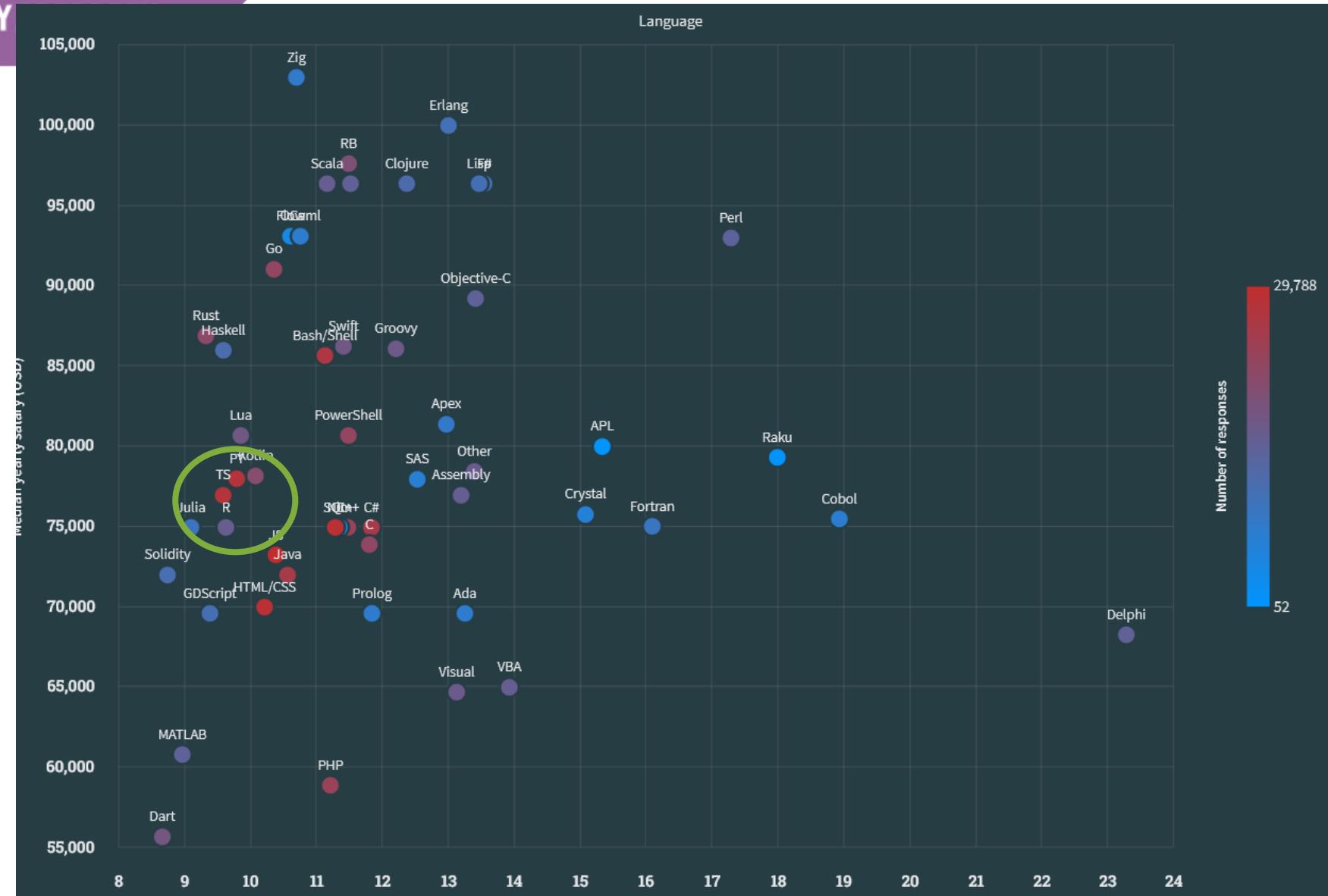
Top paying technologies (StackOverflow Survey 2023)

<https://survey.stackoverflow.co/2023/#section-top-paying-technologies-top-paying-technologies>

Top paying programming languages Survey

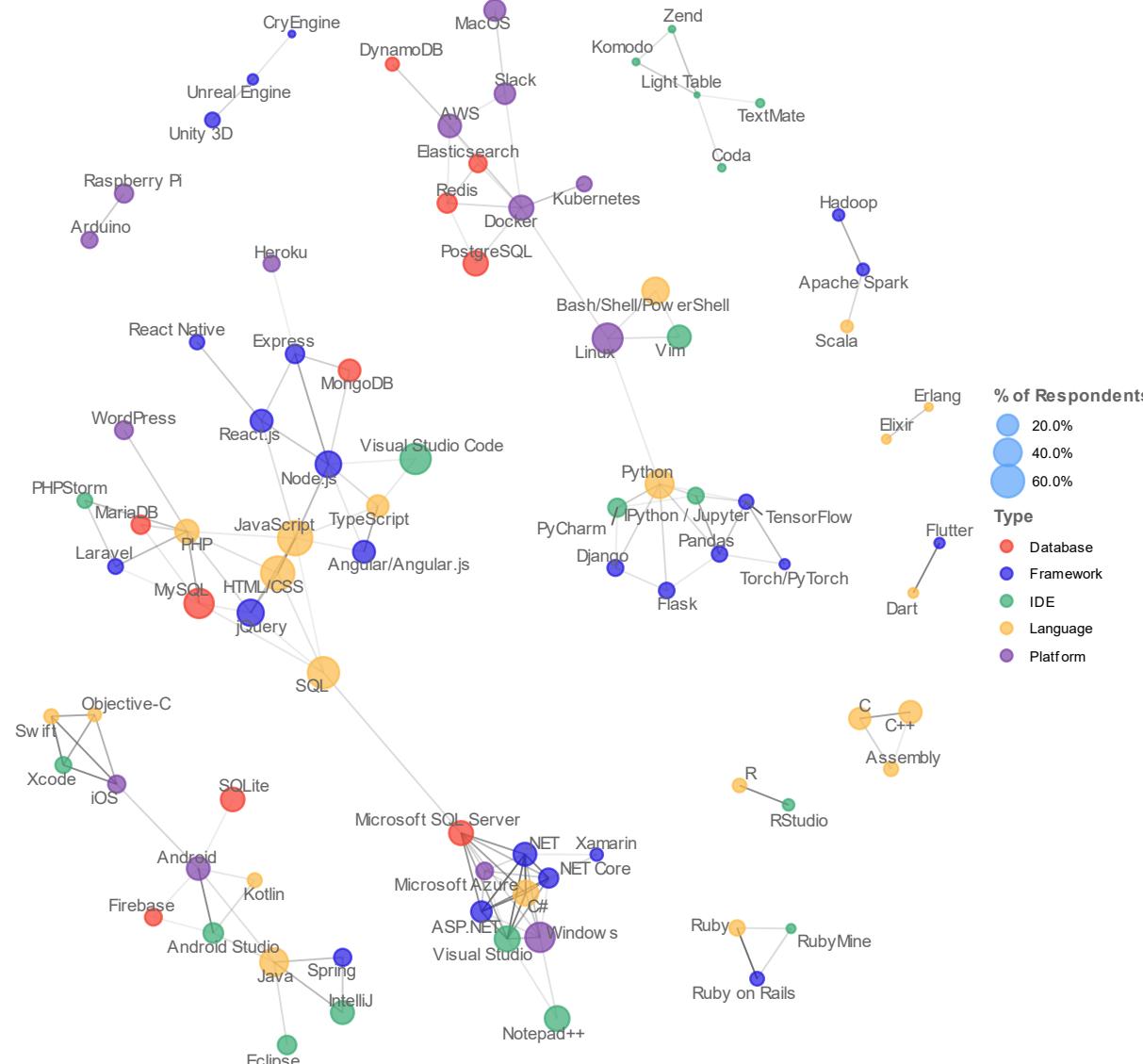
https://www.hired.com/research/salary-survey/





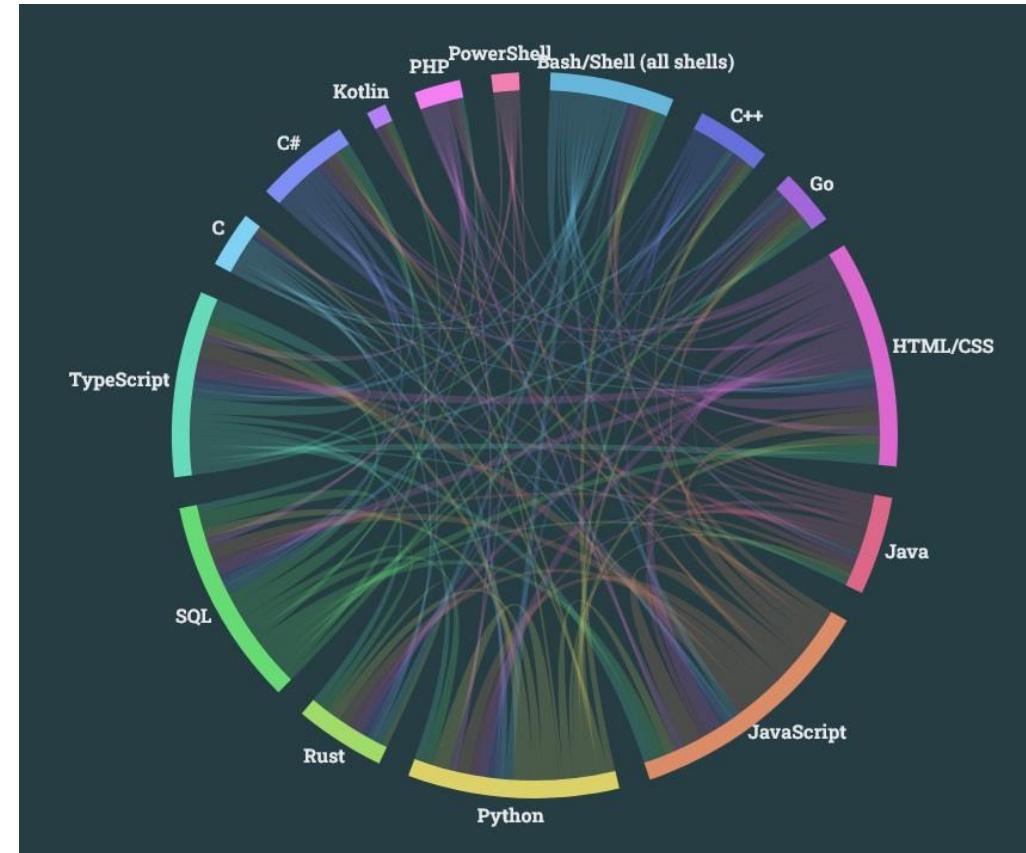
<https://survey.stackoverflow.co/2023/#section-salary-salary-and-experience-by-language>

How they connect?



<https://insights.stackoverflow.com/survey/2019>

This is close!



<https://survey.stackoverflow.co/2023/#section-worked-with-vs-want-to-work-with-integrated-development-environment>

What are R and Python?

- Open-source programming language for statistical analysis, graphics, data science and machine learning.
- Command-line based, however, complementary tools provide a friendly user interface(s).
- Will require you to learn both syntax and semantics.

What do we use them for?

- Exploratory and statistical data analysis.
- Visualisation and graphics.
- Data preparation (data wrangling).
- Machine learning and modelling.

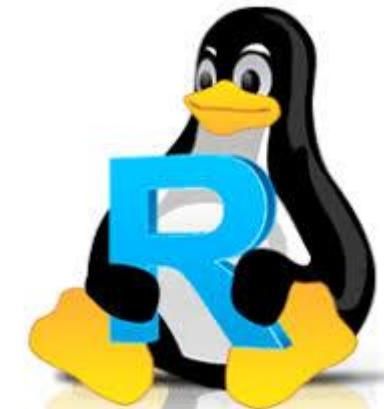


Why these tools?

- Free.
- Easy to use.
- Have packages for everything.
- Have great online support community.
- Statistics tools AND programming languages.
- Available across platforms.
- Similar to MATLAB.
- Robust for visualisations.
- You can produce reports of your work easily.



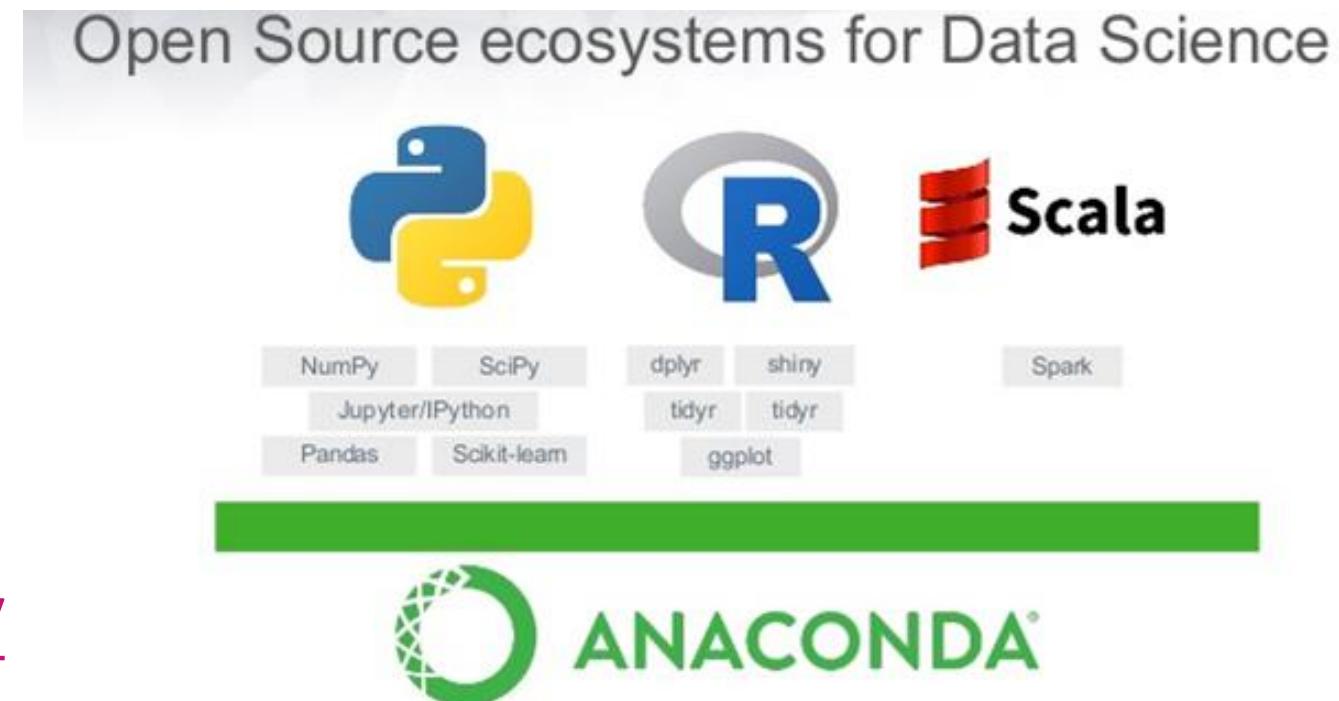
`pythonic.love()`



How to Install?

1. Download R
 - <http://cran.r-project.org/>
2. Download RStudio (IDE)
 - <http://www.rstudio.com>
3. Download Python
 - <https://www.python.org/downloads/>
4. Download an IDE
 - <https://www.spyder-ide.org/>
 - <https://www.jetbrains.com/pycharm/>
 - <https://jupyter.org/>

OR Download Anaconda!
<https://www.anaconda.com/>



Where to Learn?

- Coursera
 - <https://www.coursera.org/>
- Datacamp
 - <https://www.datacamp.com/>
- RGU offers online short courses!
 - <https://www.rgu.ac.uk/study/courses/3274-introduction-to-data-science-with-python-15-credits-at-scfq-level-9>
 - <https://www.rgu.ac.uk/study/courses/5630-computer-vision-for-the-energy-sector>

3.2. Demo of Python and R

- R in RStudio
- Python in
 - Console
 - Spyder
 - PyCharm
 - Jupyter Notebook
 - Rise Slides

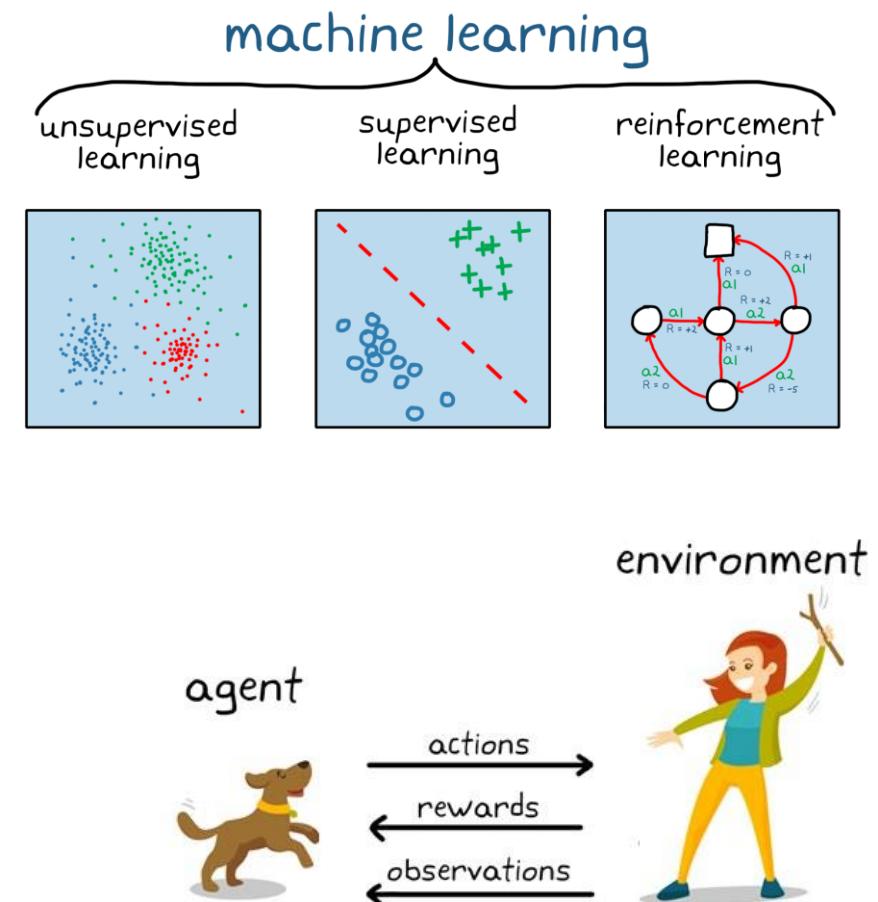
3.3. Demo of Online Notebook Platforms

- Google Colab (mostly for Python)
- Kaggle (both, plus a great data source)
- GitHub (Code repository)
- HuggingFace

4.1 Types of ML

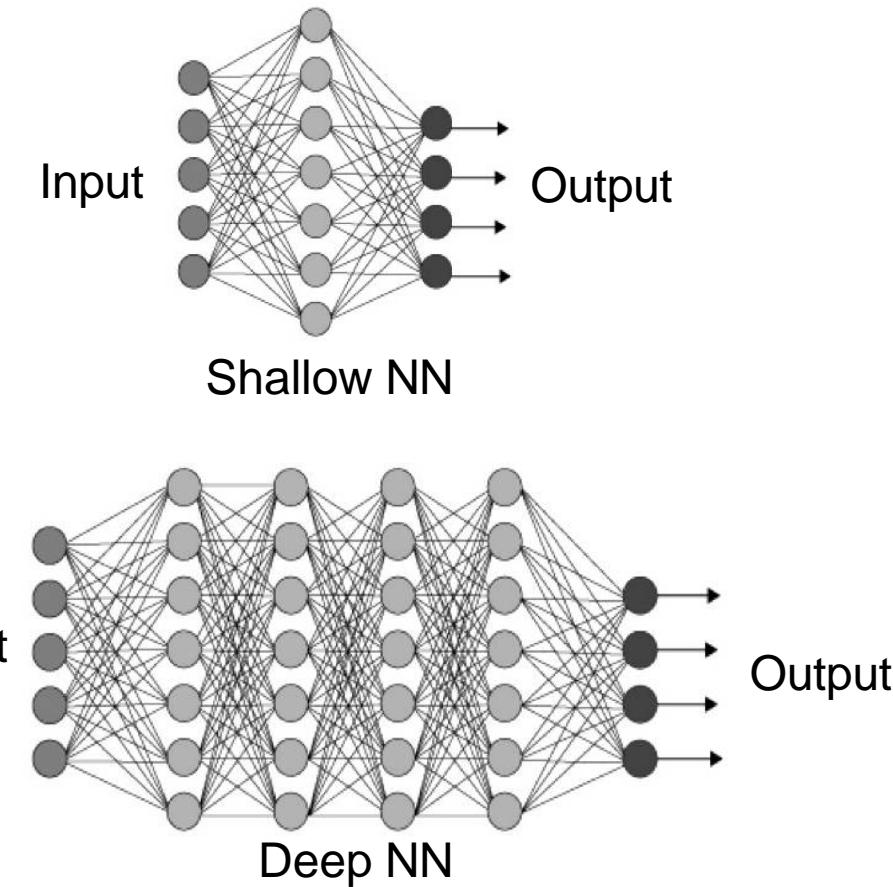
(Main) Types

- Supervised
 - Learn from labelled data, used to classify or predict
 - e.g. object recognition systems, spam detectors
- Unsupervised
 - Initial data is not labelled, insights are drawn by processing data (structure is unknown in advance), clustering, association, anomaly detection.
 - e.g. user behaviour analysis, market basket analysis
- Reinforcement
 - No supervisor, only a reward signal is used for an agent to determine if they are doing well or not
 - Type of dynamic programming, system learns from its environment, maximises the gain
 - Reward, no training data involved, learns on the go via trial and errors (rewards and punishments)
 - e.g. self driving cars to navigate through the traffic
https://www.youtube.com/watch?v=W2CAghUiofY&feature=emb_logo

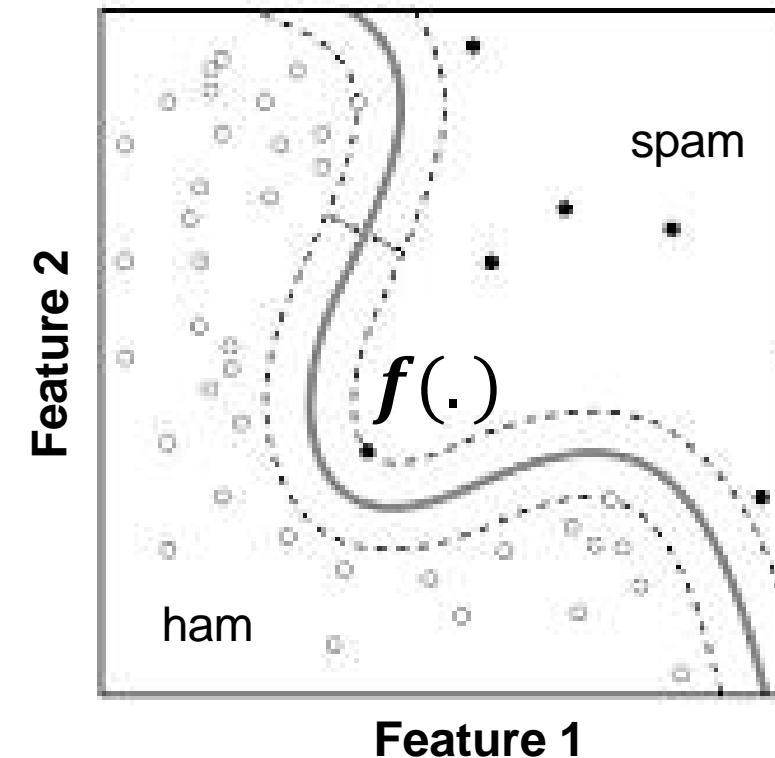
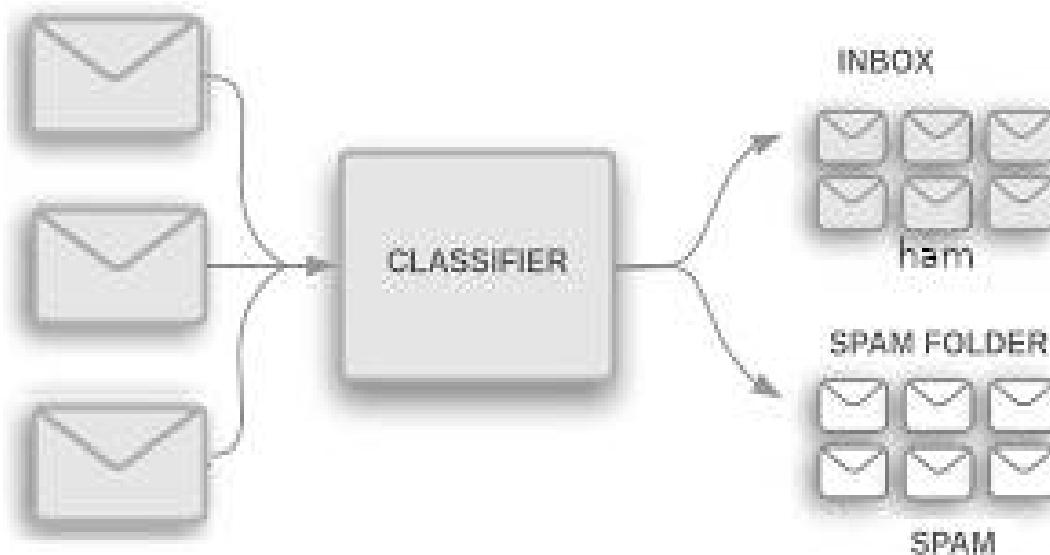


(More) Types

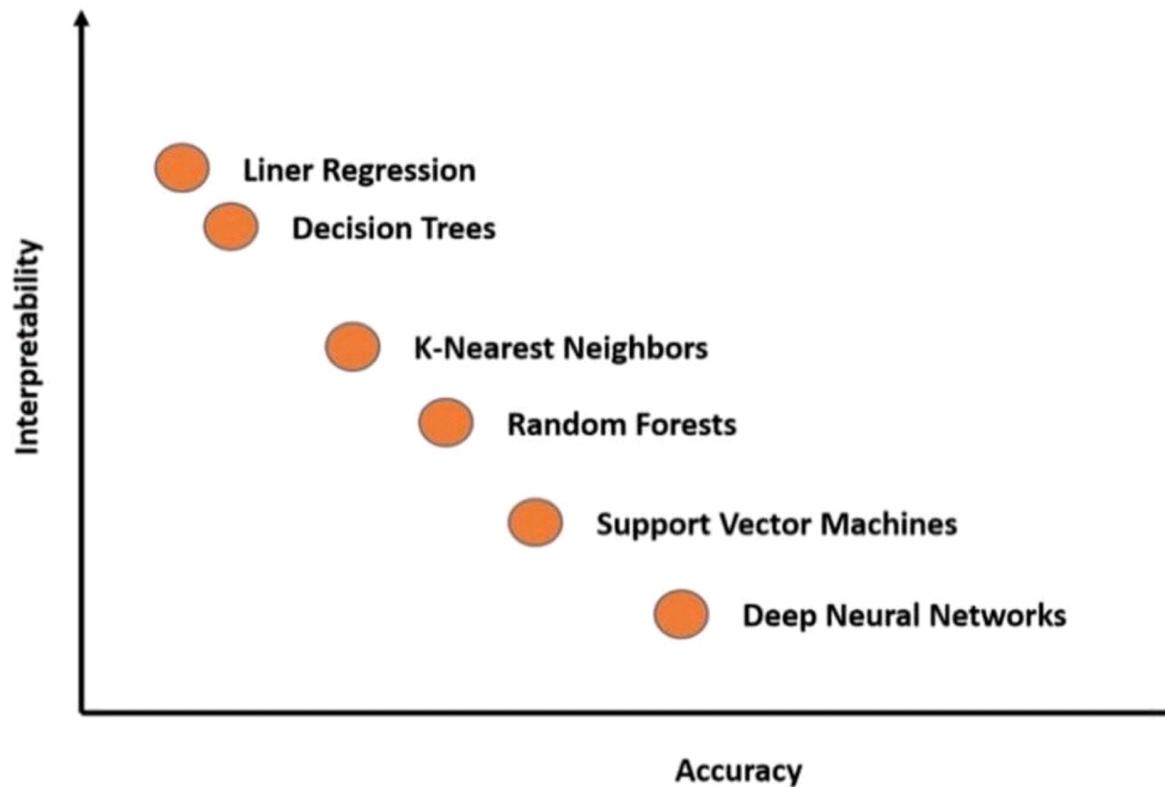
- Semi-Supervised
 - Initial training data is incomplete, both labelled and unlabelled data are used in the training
- Deep
 - Neural networks (NN) with number of hidden layers Back-propagation to train deep neural nets
- Active
 - User constantly feeds algorithm with labels
 - Used for NLP tasks (systematic reviewing)
- Imitation
 - Used mostly in robotics



Solving a problem using ML logic



Accuracy and Interpretability

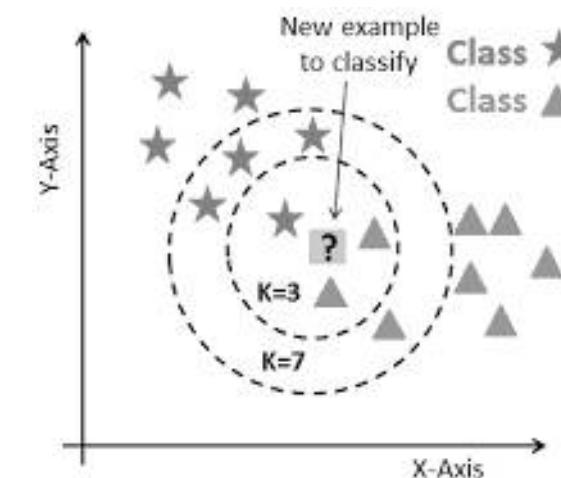


K Nearest Neighbours (KNN)

- Classified by a majority vote of neighbours
 - K is an integer specified by human
 - Non-parametric algorithm - not making any assumption on data distribution
 - Lazy algorithm - does not really learn any model and make generalisation of the data
- Advantages: Simple to implement, robust to noisy training data, and effective if training data is large
- Disadvantages: Need to determine the value of K computation cost is high

% R CODE

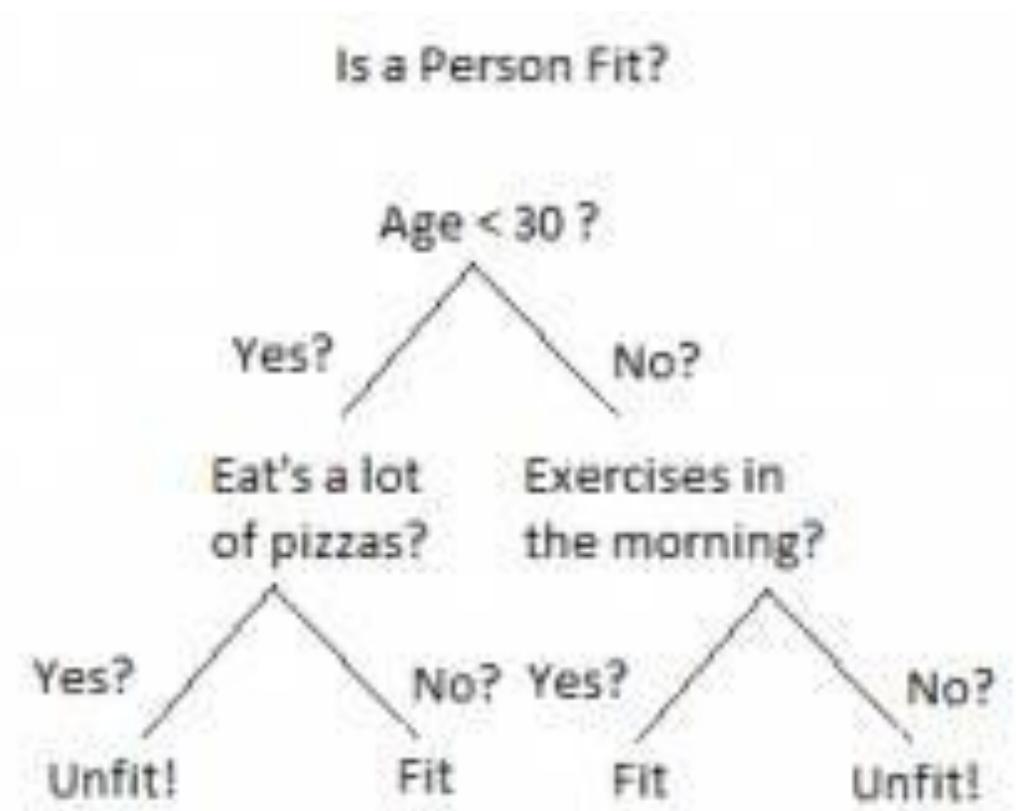
```
library(caret)
fit.knn <- train(Species~.,
                   data=dataset, method="knn")
predicted= predict(fit.knn,x_test)
```



Decision Tree/Random Forest

- Segment the predictor space into multiple regions
- Each region has only a subset of the training dataset
- High variance → Small changes in the training data can give an entirely different decision tree model
- Led to better classifiers → Random Forest

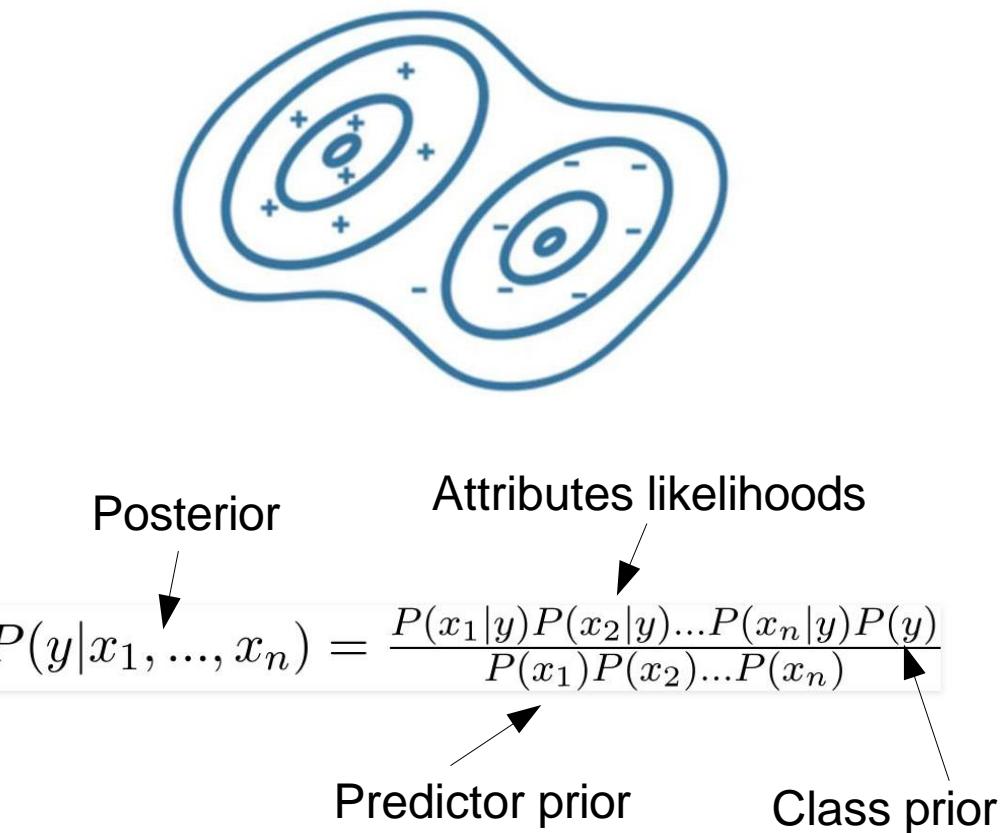
```
library(caret)
fit.dt <- train(Species~.,
                  data=dataset, method="rpart")
predicted= predict(fit.dt,x_test)
```



Naïve Bayes (NB)

- Probabilistic classifier inspired by the Bayes theorem
- Assumes attributes are conditionally independent
- Advantages: small amount of training data required, extremely fast (on training)
- Disadvantages: zero probability problem, if the conditional probability is zero for a particular attribute...
- “Hard” to understand, simple to implement

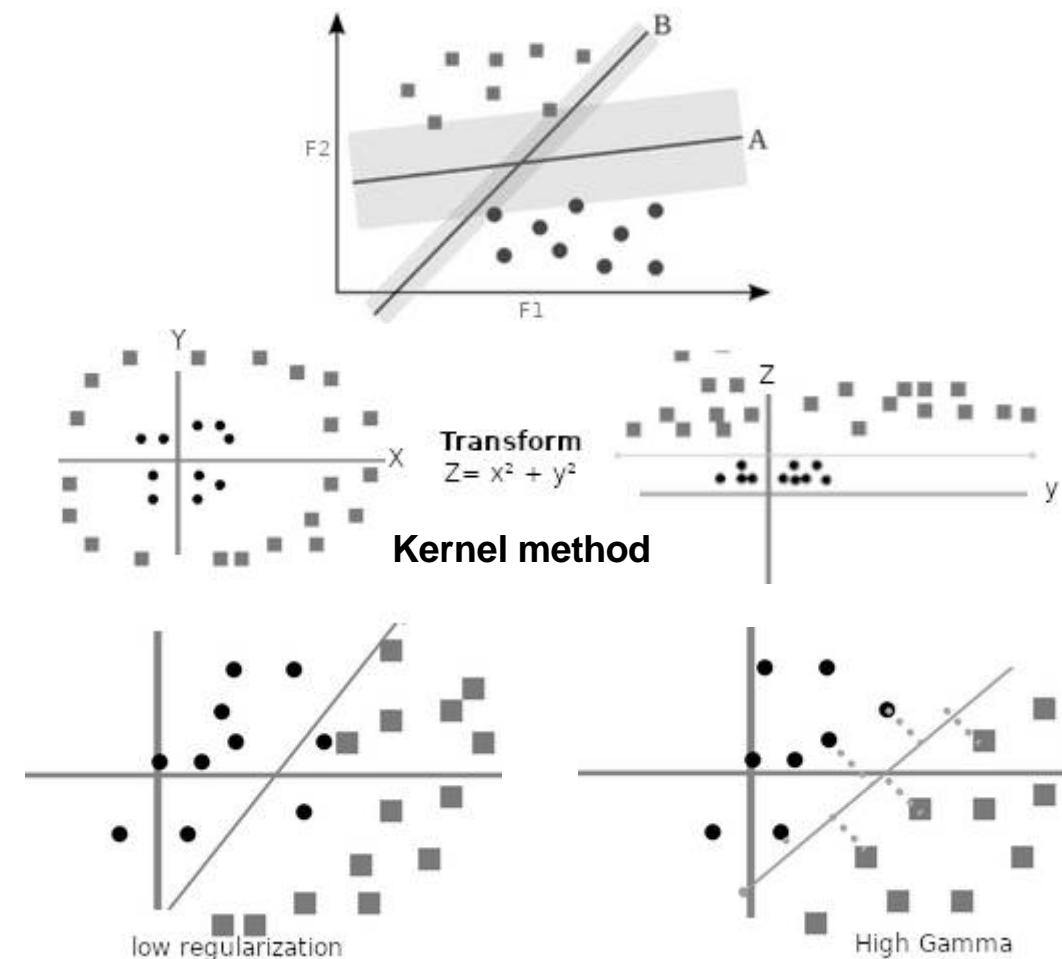
```
x <- cbind(x_train,y_train)
fit <-naiveBayes(y_train ~ ., data = x)
predicted= predict(fit,x_test)
```



Support Vector Machine

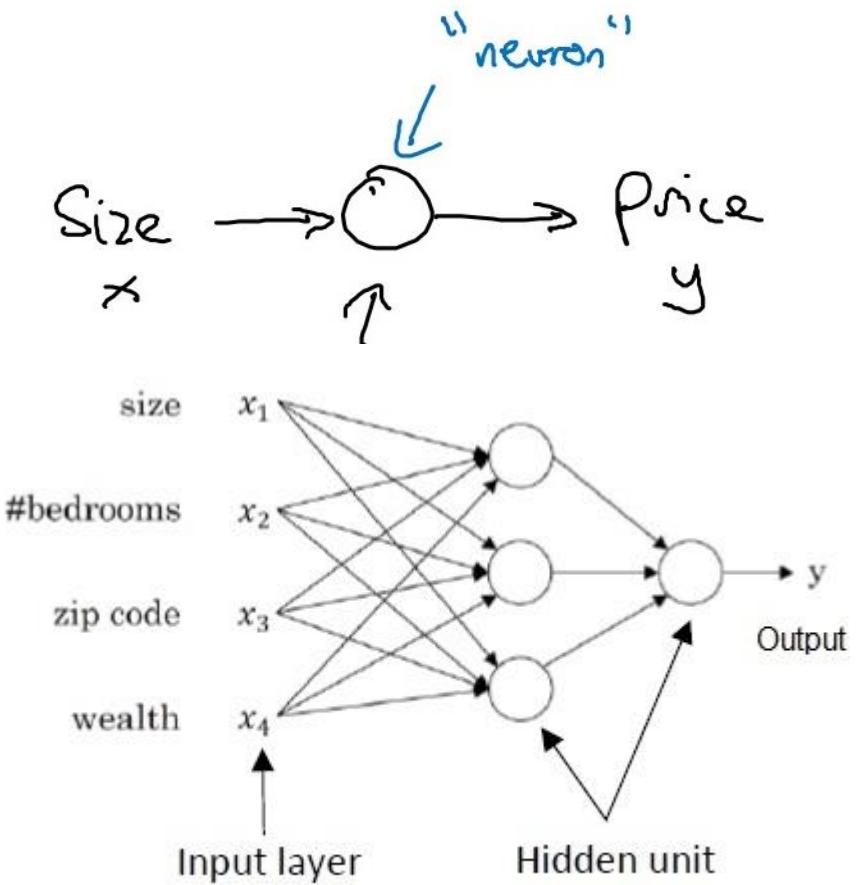
- Discriminative classifier defined by a separating hyperplane
- Tuning parameters in SVM classifier
 - Kernel - transformation method, e.g. Polynomial and exponential kernels
 - Regularization - how much to avoid misclassifying each training example
 - Gamma - how far the influence of a single training example reaches, high gamma → only nearby examples
- A margin in SVM is a separation of line to the closest class points
 - Good margin is one where this separation is larger for both the classes

```
x <- cbind(x_train,y_train)
fit <- svm(y_train ~ ., data = x)
predicted= predict(fit,x_test)
```



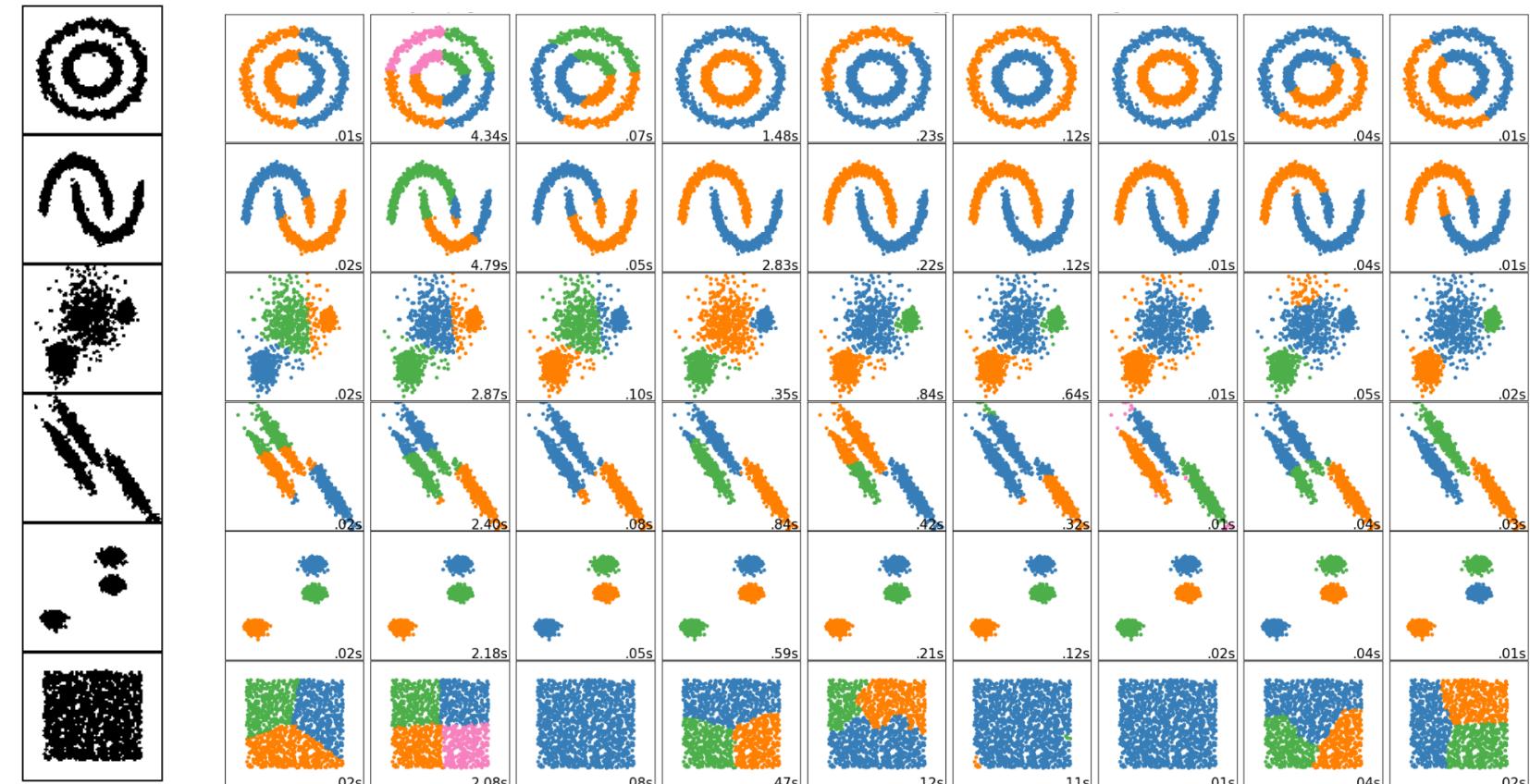
Neural Network (NN)

- Each “neuron” is trained to be activated/deactivated given certain weights and biases
- Multiple neurons work together to solve a multi-variable problem
- If more layers are added to the NN model, second-order relations can be discovered
- There are “easy models” that can be implemented in three lines of code, but if you want to do it properly, you need to be more skilled!

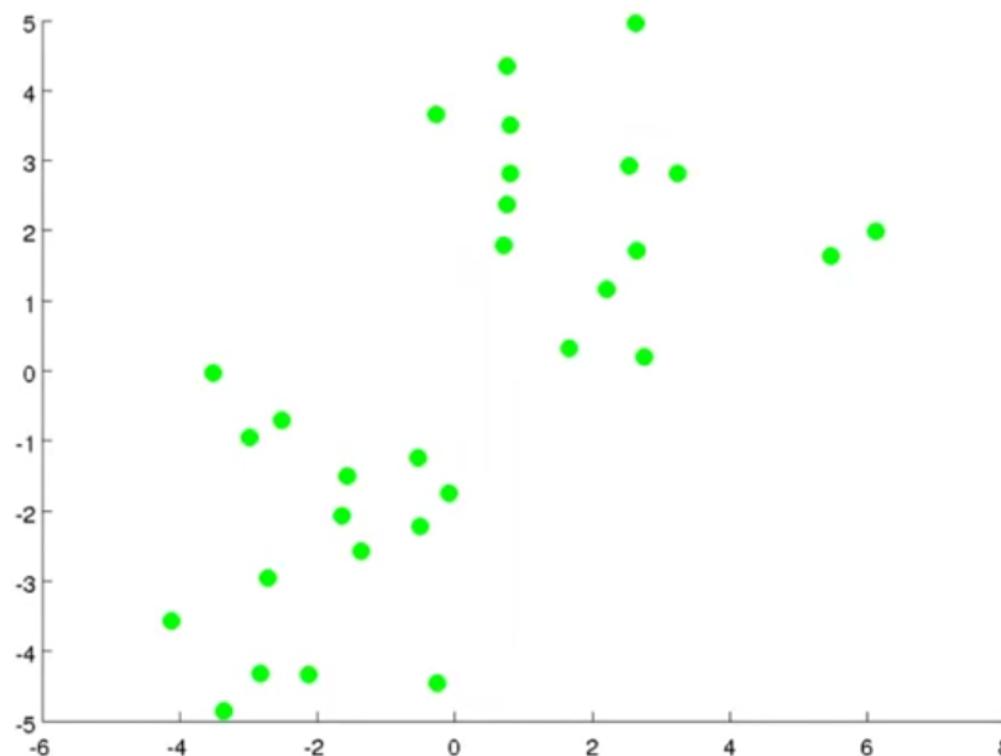


Clustering (unsupervised) with K Means

- Not to be confused with KNN!
- Two main steps:
 1. Cluster assignment
 2. Centroid recalculation

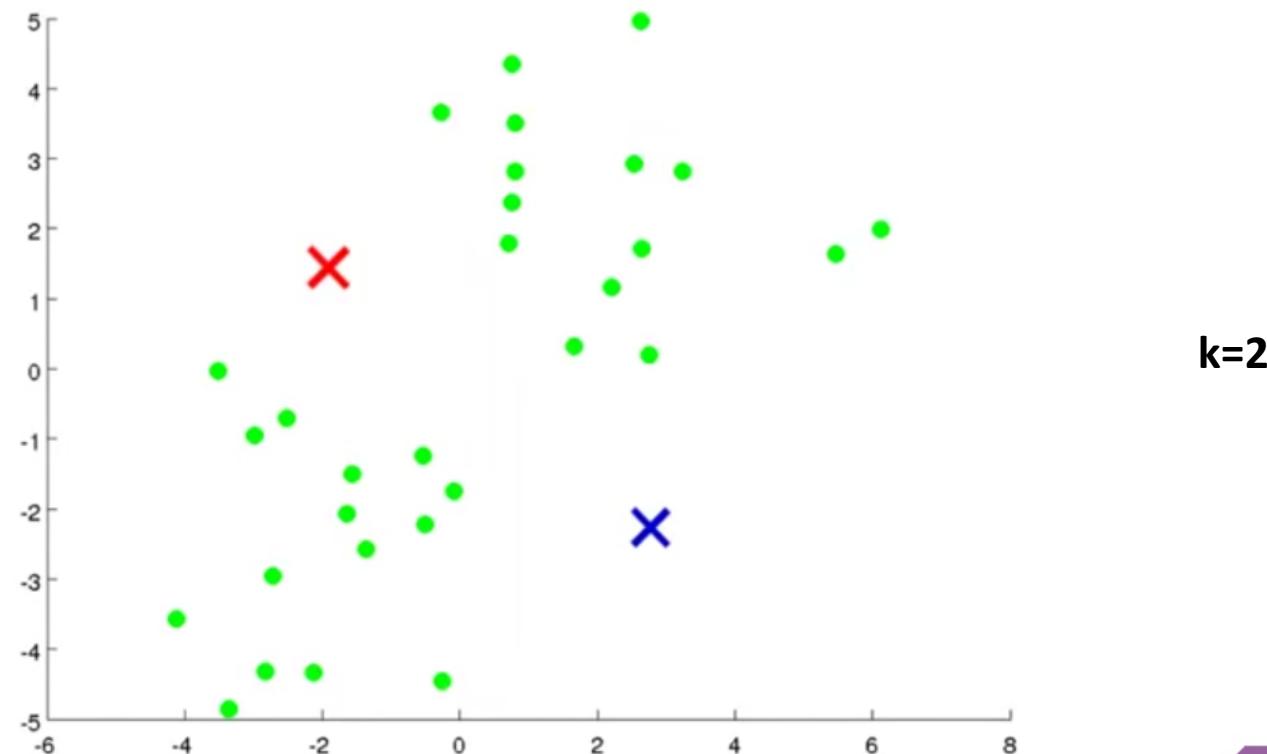


Example



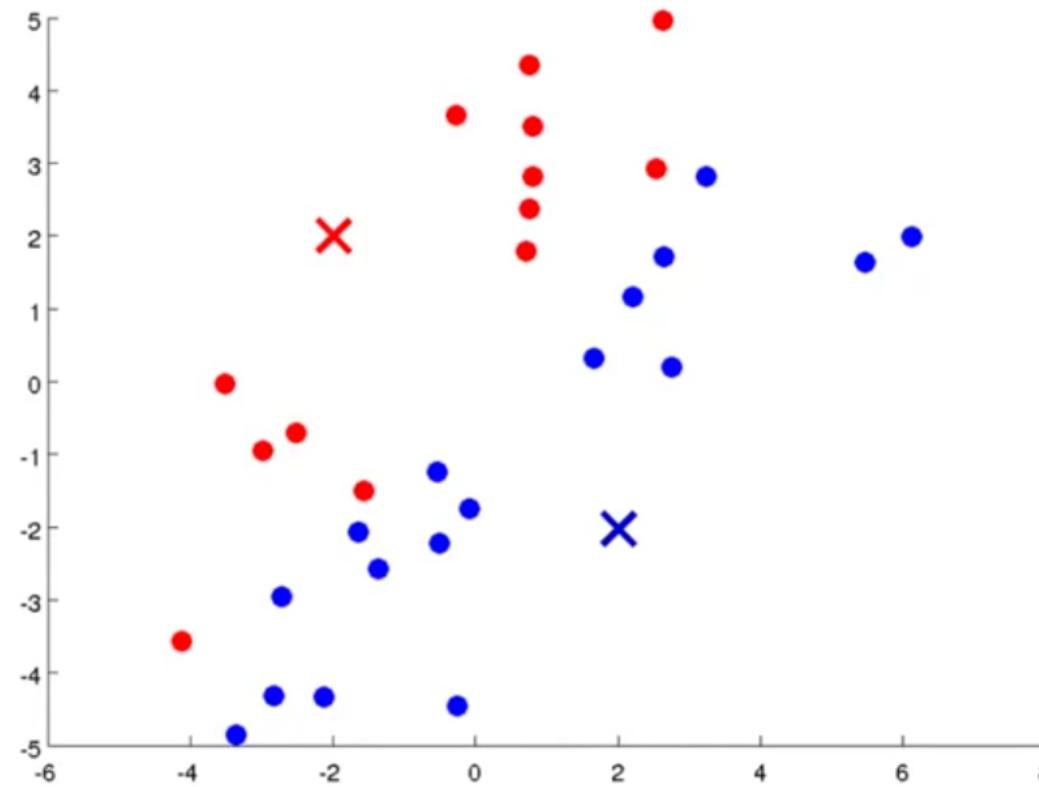
<https://www.youtube.com/watch?v=Ao2vnheIKhi>

1. Select the number of clusters & randomly initialise k centroids.



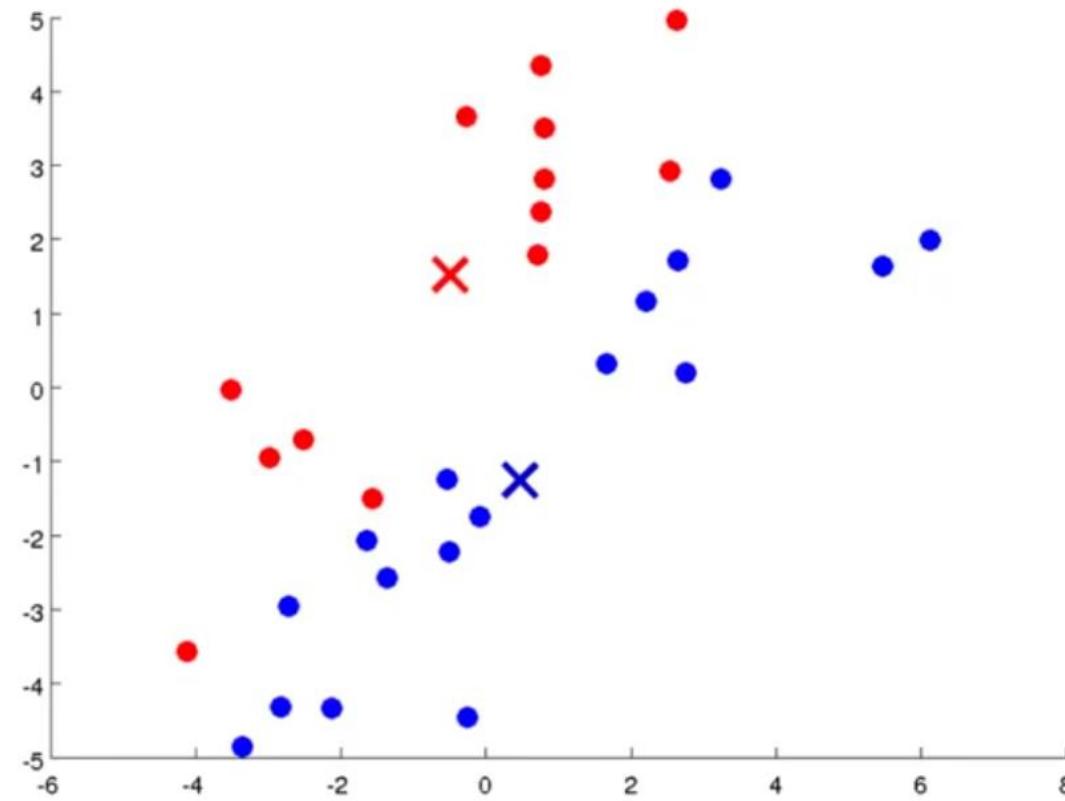
<https://www.youtube.com/watch?v=Ao2vnheIKhi>

2. Assign each data point to a cluster (red or blue) according to the minimum distance to a centroid.



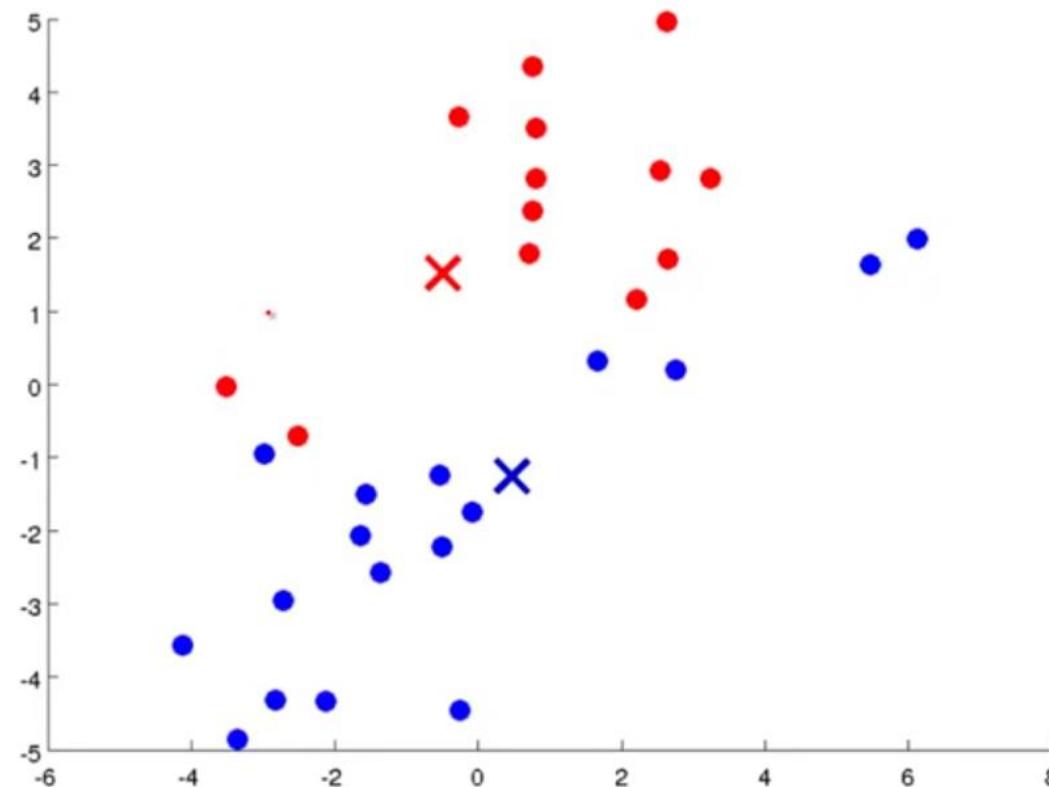
<https://www.youtube.com/watch?v=Ao2vnheIKhi>

3. Compute the mean of each cluster and *move the centroid* to that position.



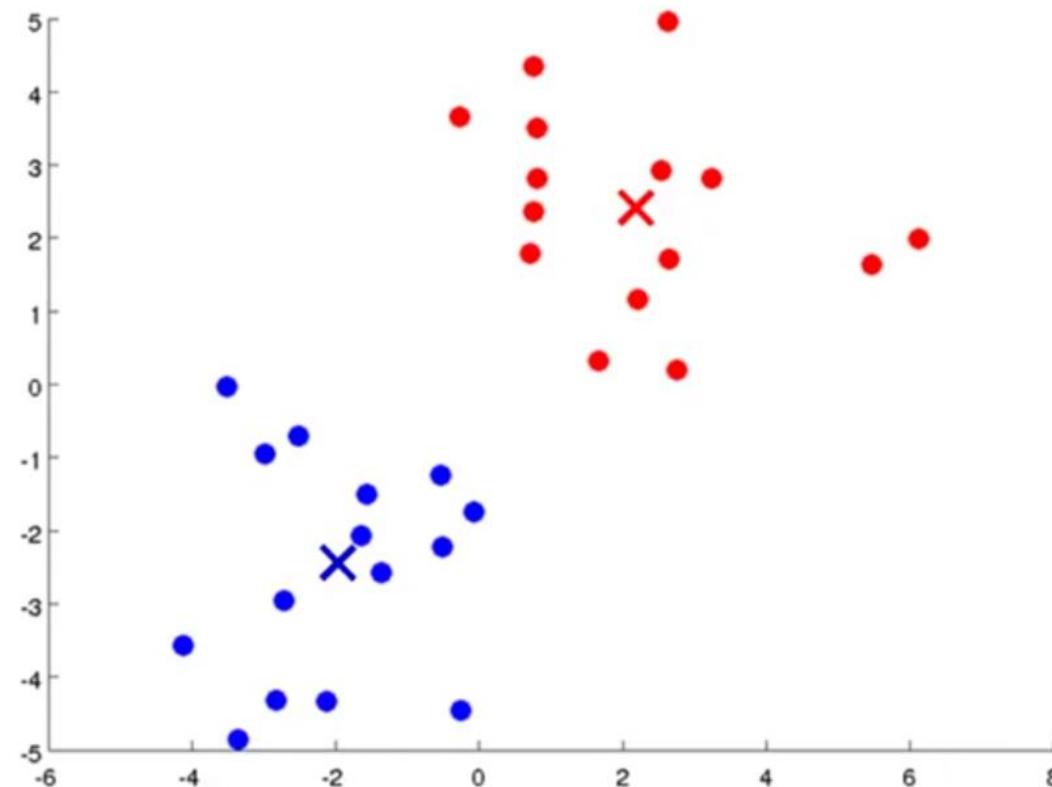
<https://www.youtube.com/watch?v=Ao2vnheIKhI>

4. Compute a new set of clusters based on the new centroids.



<https://www.youtube.com/watch?v=Ao2vnheIKhi>

5. Iterate until clusters don't change (convergence).



<https://www.youtube.com/watch?v=Ao2vnheIKhi>

4.2 Practical Example (Iris)

Link to online Jupyter Notebook:

<https://colab.research.google.com/drive/1gDVOrqs3d3Ycb7X2wY7XC9oCMsKzb0j7?usp=sharing>

5. Evaluating ML

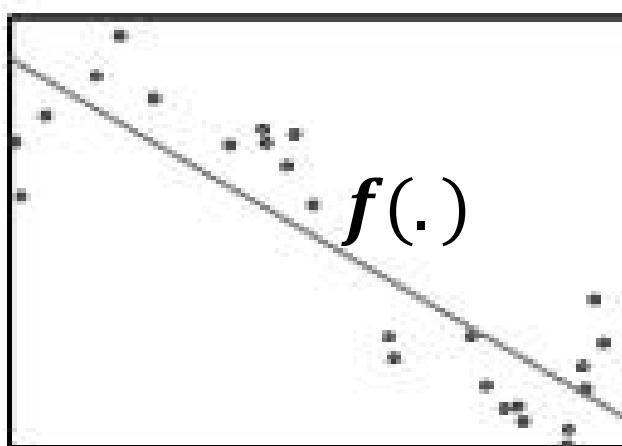
Evaluating ML

- Now that we know how to implement (basic) ML algorithms, we need to assess their effectiveness
- Accuracy is **not** the only way!
- There are numerous metrics that help us understand how well our algorithms are performing

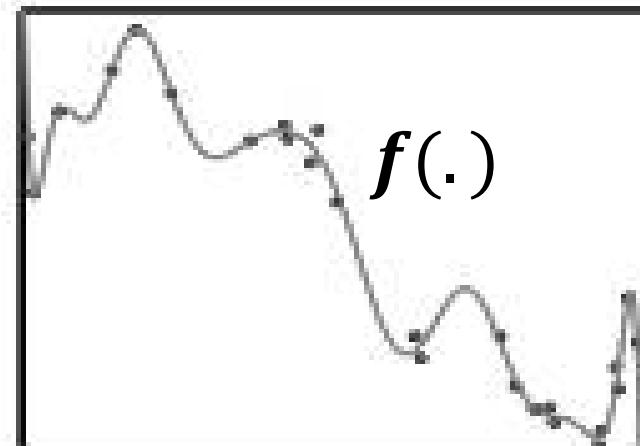


Over/Under Fitting

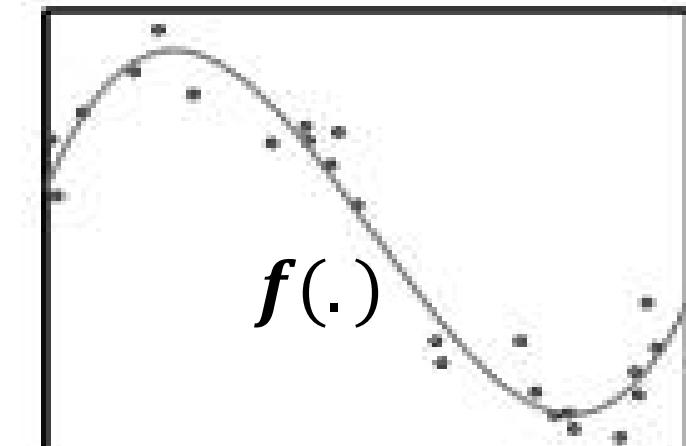
- Assume we want to fit a regression to a series of observations:



Underfitting



Overfitting



Good fit

Assuming a binary scenario

- True Positives (TP)
 - This is what many people think accuracy is (but it's not!)
 - Samples from the positive class that are classified correctly
- True Negatives (TN)
 - How many samples from the negative class are **NOT** classified as being from the positive one
- False Positives (FP)
 - How many samples from the negative class are classified as being from the positive class
 - Also known in statistics as **False Alarms** or **Type I Error**
- False Negatives (FN)
 - How many samples from the positive class are classified as being from the negative class
 - Also known in statistics as **Type II Error**

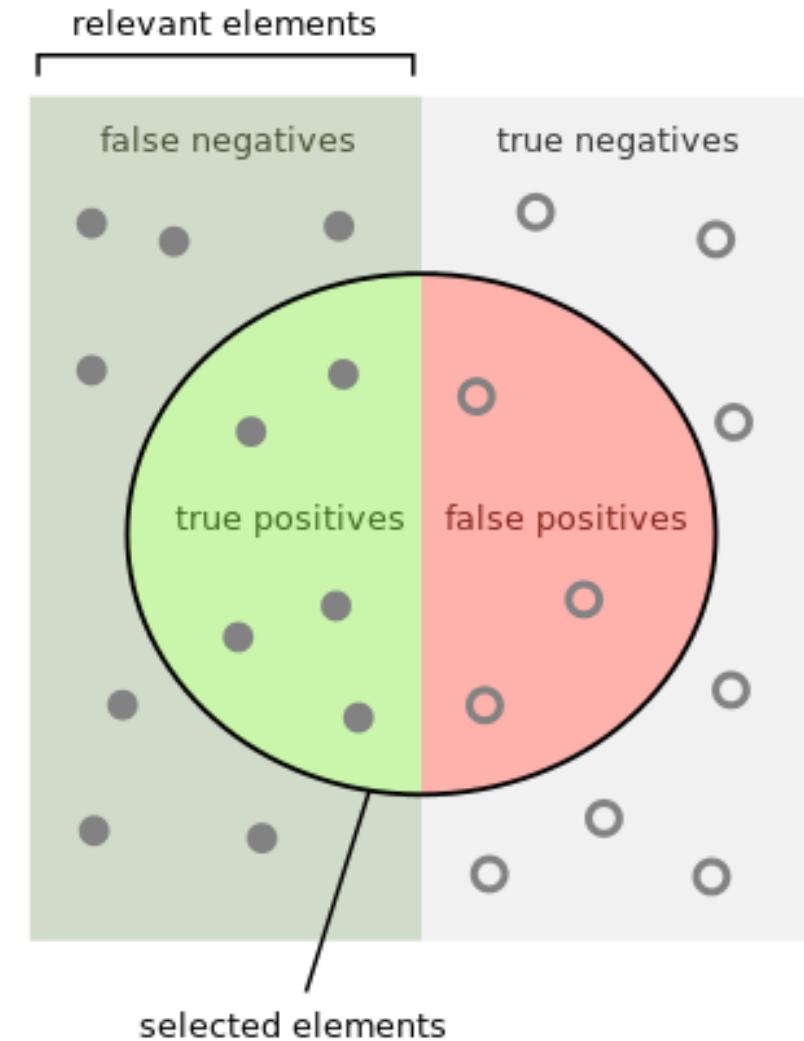
So what is the accuracy?

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Value between 0 and 1
- Not a suitable metric for **imbalanced** scenarios
- Why?

Precision and Recall

- These metrics focus on the balance between the relevant and irrelevant elements
- Consider if the positive case is easy to find

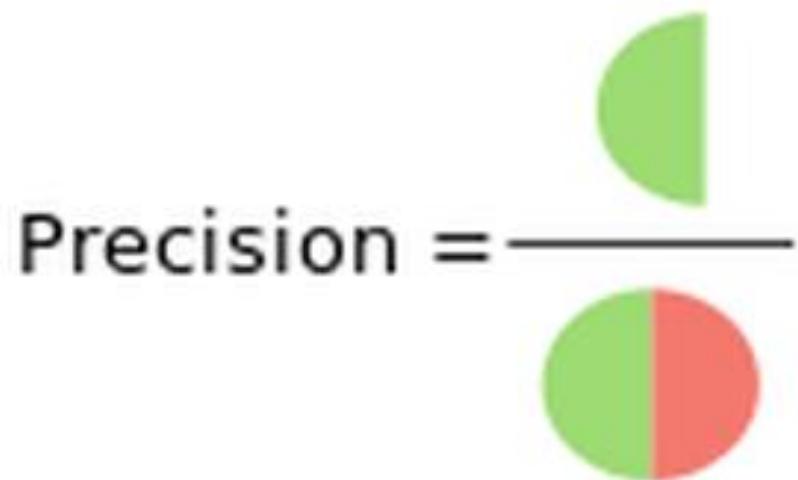


Precision

How many selected items are relevant?

$$Precision = \frac{TP}{TP + FP}$$

- How much of what I **have** I **need**?



Recall

How many relevant items are selected?

$$Recall = \frac{TP}{TP + FN}$$

- How much of what I **need** I **have**?



F1-score

- Harmonic mean between precision and recall
- Helps gauge both
- There are different interpretations and variations

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{(2 \times TP) + FP + FN}$$

Confusion Matrix

		Actual class	
		Cat	Dog
Predicted class	Cat	5	2
	Dog	3	3

Confusion Matrix

		Actual class	
		Cat	Non-cat
Predicted class	Cat	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	3 True Negatives

Multiclass

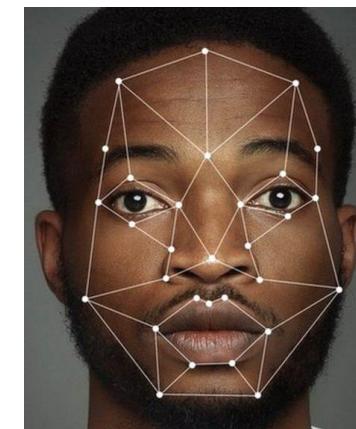
- There are many ways to adapt the aforementioned metrics to these scenarios, the most common one being the **One vs All** approach
 - Comparing a metric of one class against the rest as if these were a single class
- Considering that you can still calculate precision, recall and F1-score for each class (against the rest), another commonly used approach is macro/weighted/micro metrics:
 - Macro: Arithmetic mean
 - Weighted: multiply each by number of sample
 - Micro: Harmonic mean → accuracy

6. Final Considerations

Final Considerations

- Runtime
 - Not very academic, but **HUGELY** important in practice!
- How “green” is my algorithm?
 - <https://medium.com/codex/what-are-the-greenest-programming-languages-e738774b1957>
- How “ethical” is my algorithm
 - Data privacy
 - Jobs that it will take/create?
- Check that our algorithms are NOT racist!
 - <https://sitn.hms.harvard.edu/flash/2020/racial-discrimination-in-face-recognition-technology/>
- End of days!
 - ChatGPT
 - Dall-E

```
1 import time
2
3 t = time.perf_counter()
4 # do stuff
5 x=0
6 for i in range(1000):
7     x=x+i
8 # stuff has finished
9 print('Elapsed time: ',time.perf_counter() - t)
```



“Homework”

- Review the classical concepts of programming
 - Data structures (string, int, Boolean, etc)
 - Conditional (if/else, while, for)
 - Creating functions
- Get familiar with Python and/or R
 - Do a course according to your level
 - Practice!
 - https://www.w3schools.com/python/python_exercises.asp
 - <https://projecteuler.net/>