Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
Università di Pisa

Machine Learning Basics

Maria Elisabetta Pagnano & Claudia Ferraro – PhD Student in Bioengineering, Applied Sciences and Intelligent Systems at University Campus Bio-Medico of Rome

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

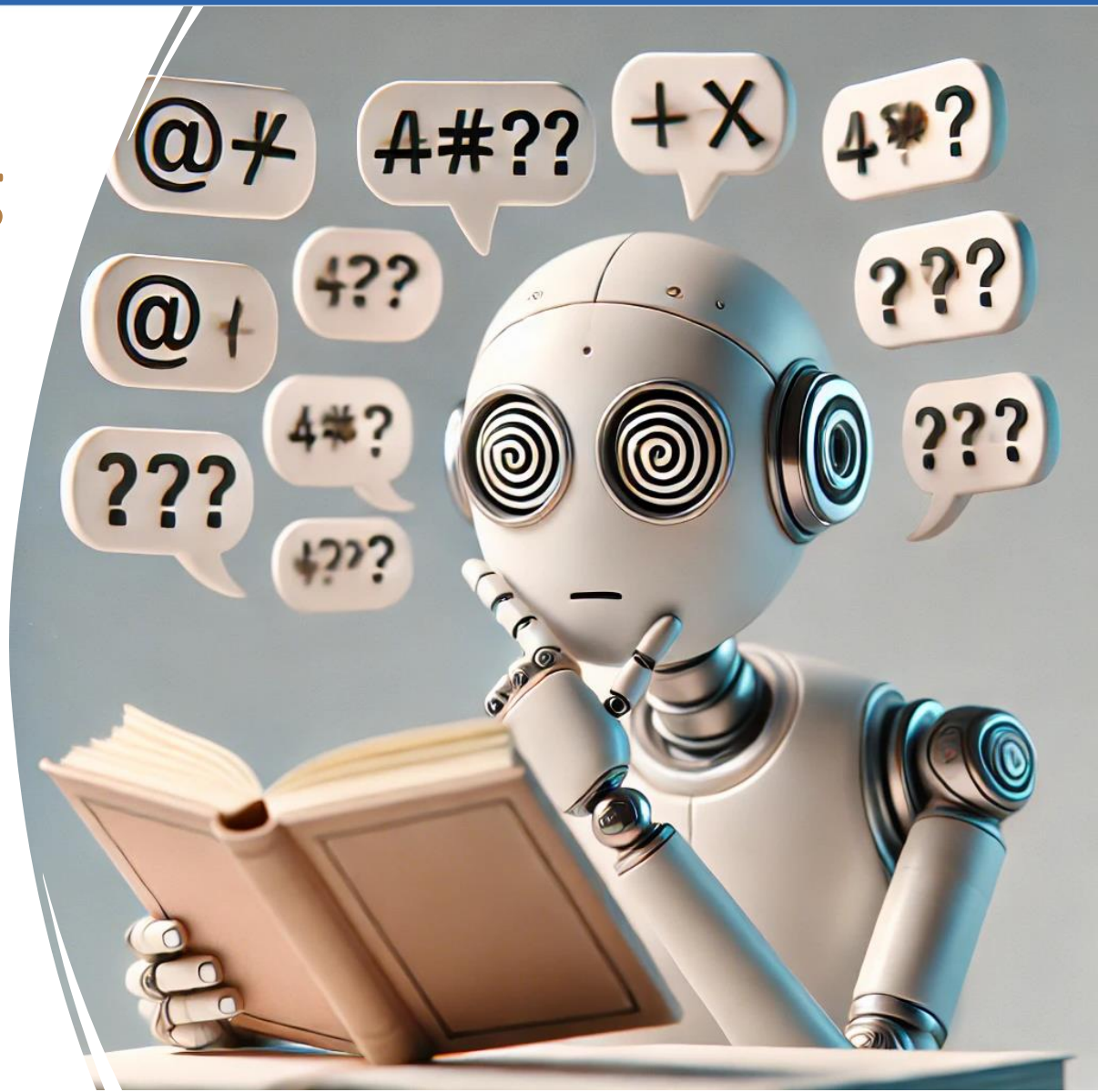
What is Machine Learning?

To learn, or not to learn.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

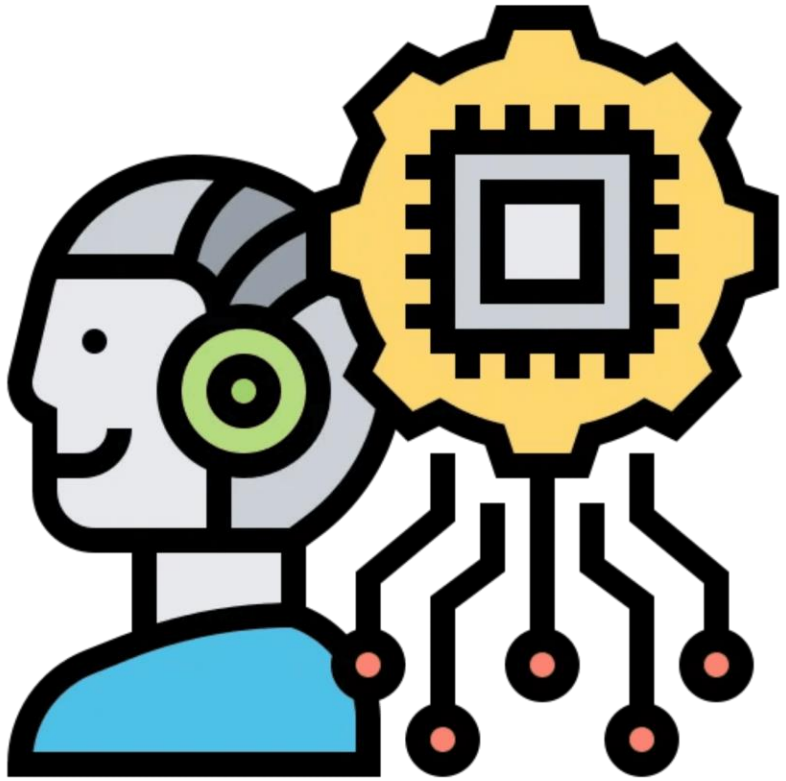
Definition of Machine Learning (Tom M. Mitchell, 1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.



Machine Learning is a field of Artificial Intelligence that allows computers to learn from data, without being explicitly programmed.

Instead of writing rules, we provide examples and the model learns to perform a task.



Healthcare: Assisted diagnosis and medical imaging analysis

Agriculture: Crop monitoring and yield prediction

Translation: Real-time language translation services

Transportation: Autonomous vehicle navigation

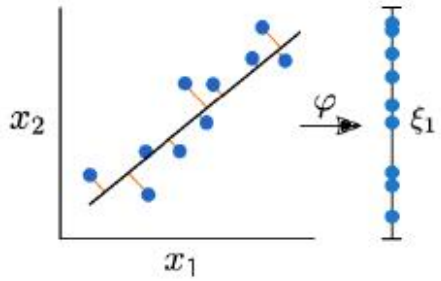
Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

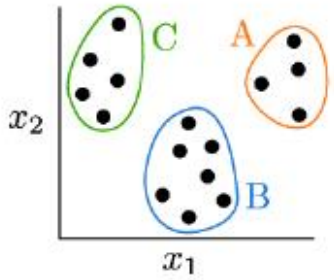
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

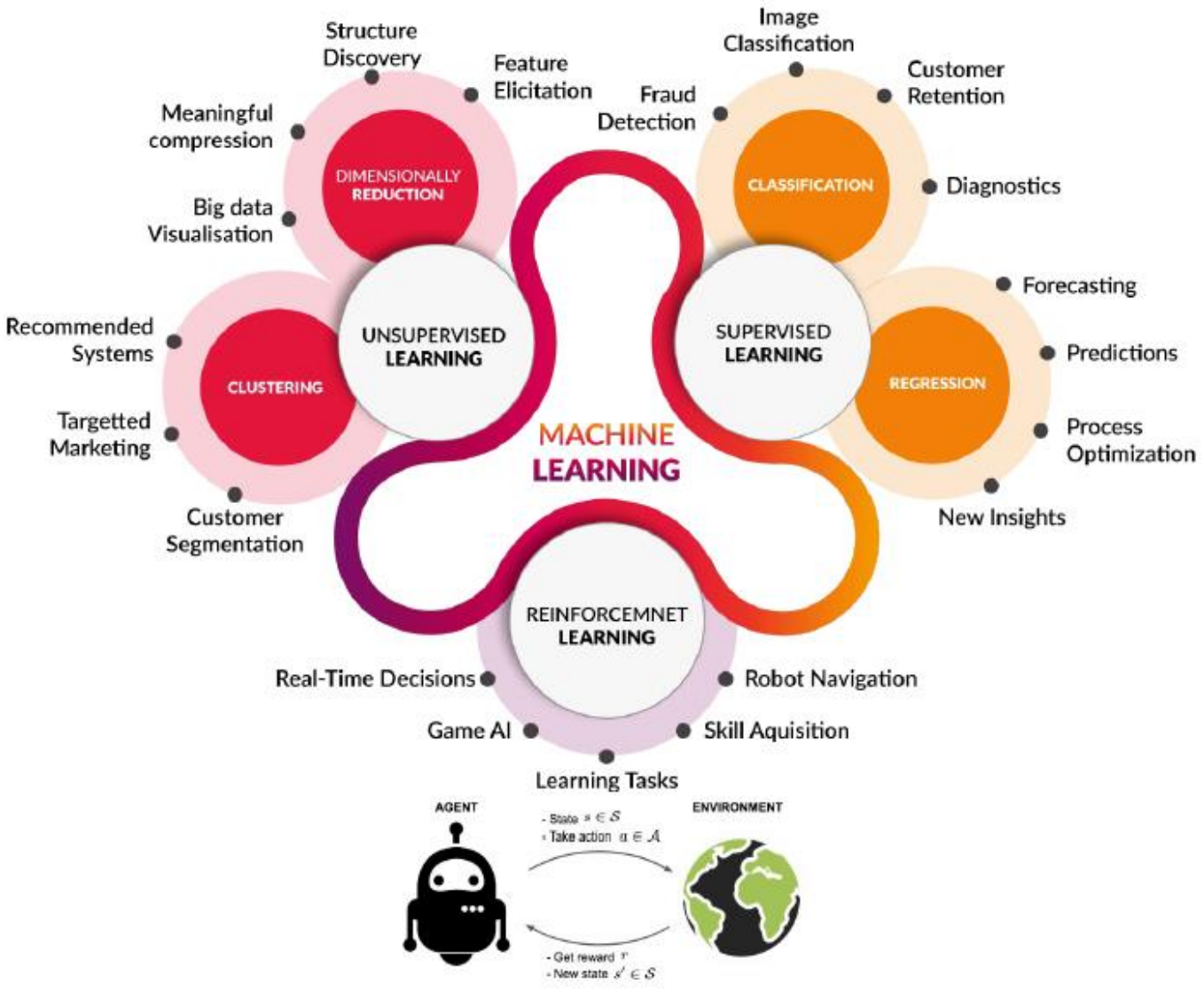
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

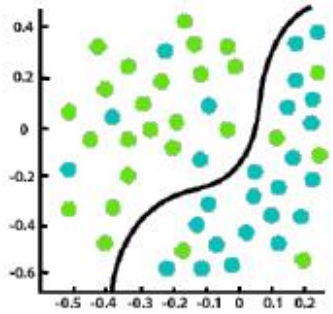
Dimensionality Reduction



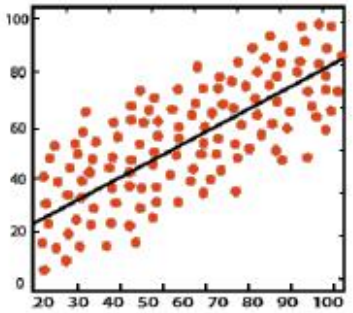
Clustering





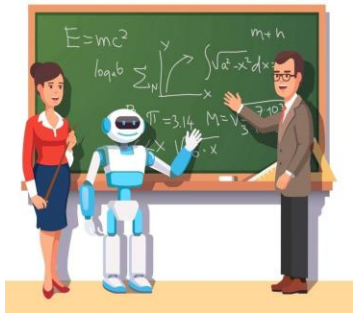


Classification



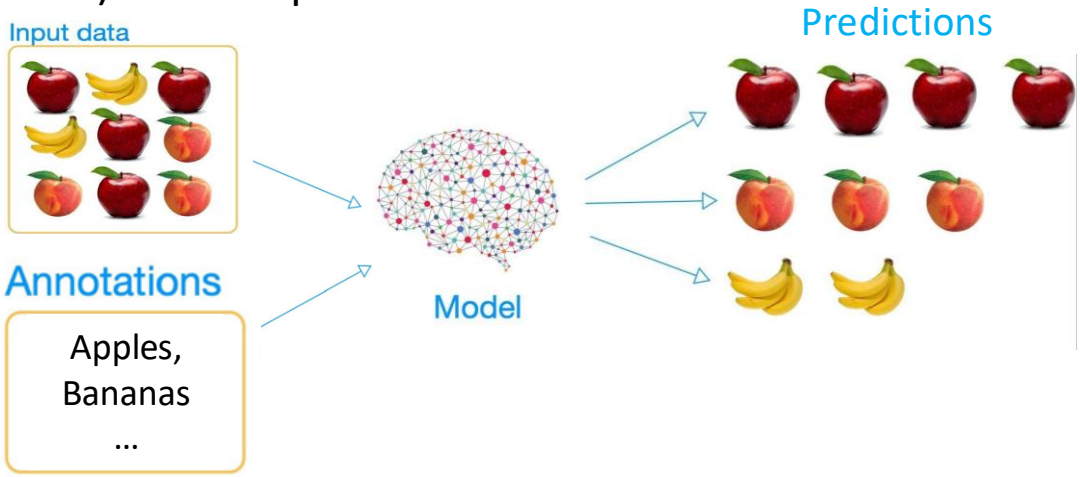
Regression

Supervised Learning: learning from **labeled** data (input and output are known) to make predictions.



Classification - The model predicts a discrete label such as dog, cat, or apple

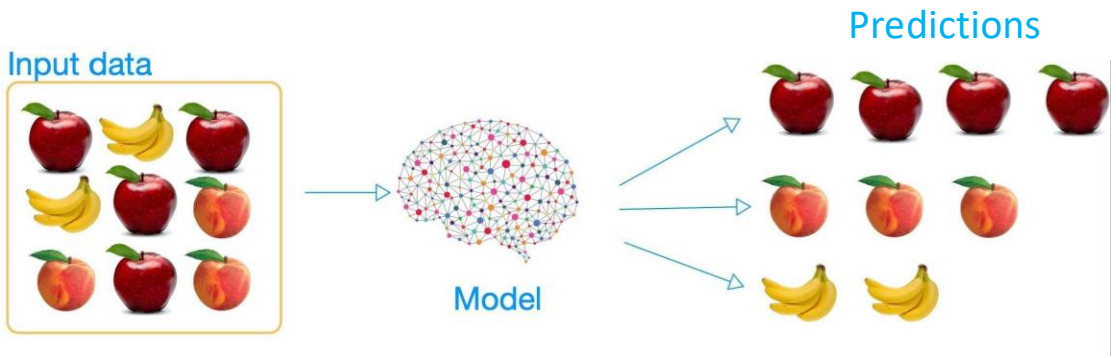
Regression - The model predicts a continuous numerical value. The goal is to estimate a number.



Unsupervised Learning: learning from **unlabeled** data to find hidden structure.

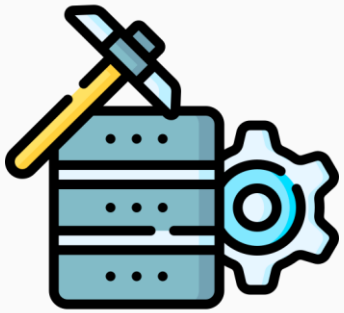
The main task is to find hidden structure in the data by grouping similar data points into "**clusters**".

- There are no "right" answers or labels to learn from.
- The algorithm finds similarities autonomously.



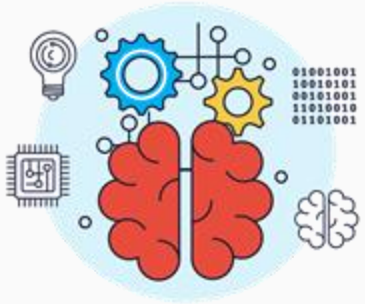
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling



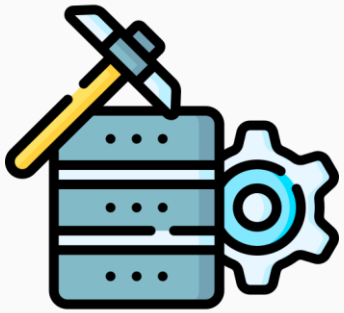
- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

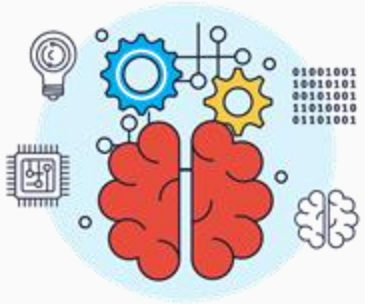
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling



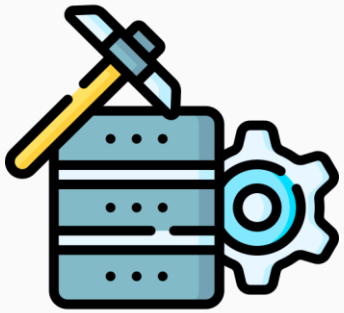
- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

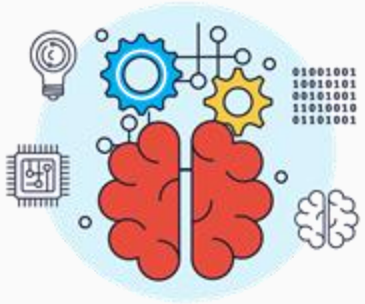
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling



- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

Data Pre-processing – Training & Test Split

We want to predict heart health:  ~ 

Independent variable (y):
Heart Health

Dependent variables (X):
Age (X1), Smoking (X2), Physical activity (X3)

Data Pre-processing – Training & Test Split

We want to predict heart health:

Independent variable (y):
Heart Health

Dependent variables (X):
Age (X1), Smoking (X2), Physical activity (X3)

Data Pre-processing – Training & Test Split

We want to predict heart health:

Independent variable (y):
Heart Health

Dependent variables (X):
Age (X1), Smoking (X2), Physical activity (X3)

Train 80%

Test 20%

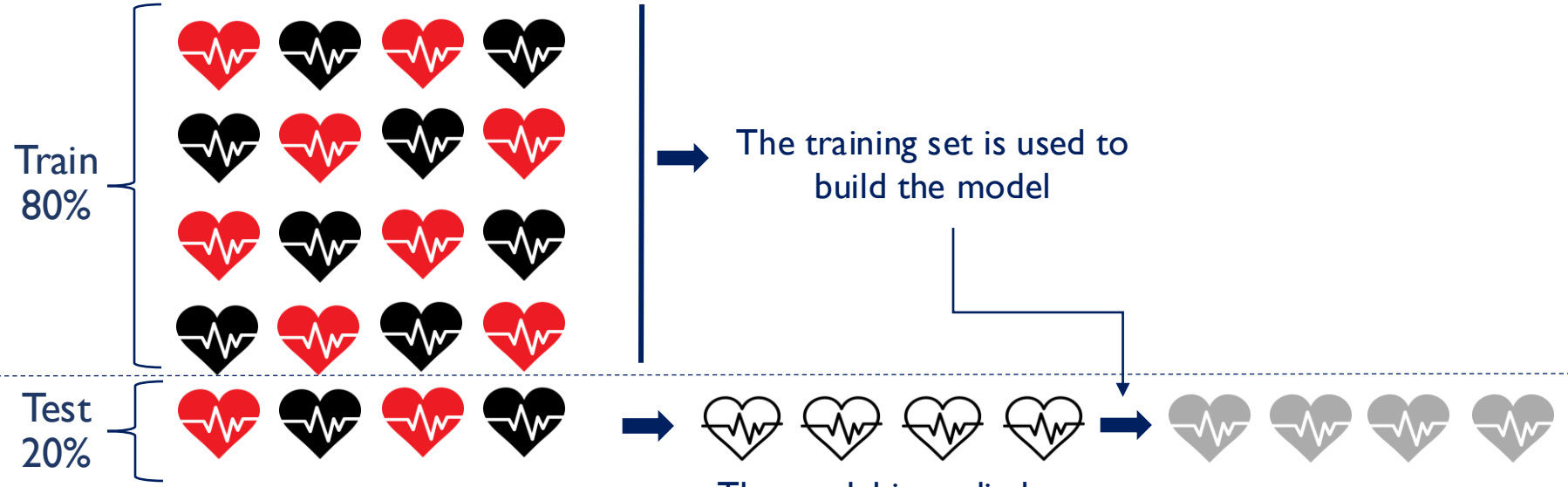
→ The training set is used to build the model

Data Pre-processing – Training & Test Split

We want to predict heart health:

Independent variable (y):
Heart Health

Dependent variables (X):
Age (X1), Smoking (X2), Physical activity (X3)



Data Pre-processing – Training & Test Split

We want to predict heart health: ~

Independent variable (y): Heart Health

Dependent variables (X): Age (X1), Smoking (X2), Physical activity (X3)

Train 80%

Test 20%

The training set is used to build the model

Model evaluation

The model is applied to the test set

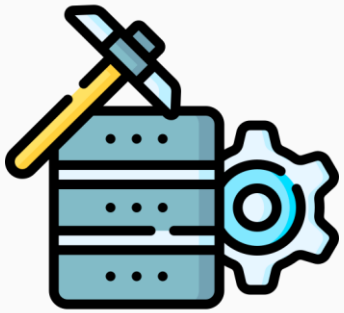
Predicted values

V.S.

True values

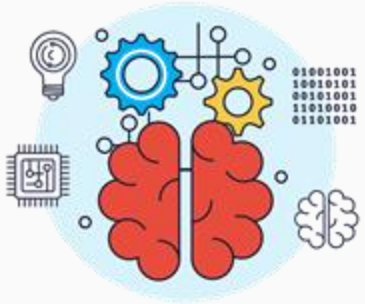
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling



- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

Data Pre-processing – Feature Scaling

Many algorithms (such as SVM, K-Means) are sensitive to feature scale. A feature with a large range (e.g., 0-100,000) can "dominate" one with a small range (e.g., 0-1).

Average price in USD	Total Sold	Smal Avocados Sold	Large Avocados Sold	Extra Large Avocados Sold
1.29	319746	292097	27351	298
1.38	272580	251774	20702	103
1.26	356227	324932	31019	276
1.37	306947	283024	23741	182
1.29	279486	250289	28890	308
1.3	300032	269799	29732	501
1.43	292814	263916	28442	456
1.42	285413	250441	34483	488
1.29	353690	290458	62980	253
1.39	301727	236814	64608	304

- **Objective**: Bring all features to a comparable scale.
- Improves the training speed (**convergence**) of the model.
- **Prevents** some **features** from **being weighted more than others** simply because of their range.
- *Essential for distance-based algorithms and neural networks.*

Data Pre-processing – Feature Scaling

Average price in USD	Total Sold	Smal Avocados Sold	Large Avocados Sold	Extra Large Avocados Sold
1.29	319746	292097	27351	298
1.38	272580	251774	20702	103
1.26	356227	324932	31019	276
1.37	306947	283024	23741	182
1.29	279486	250289	28890	308
1.3	300032	269799	29732	501
1.43	292814	263916	28442	456
1.42	285413	250441	34483	488
1.29	353690	290458	62980	253
1.39	301727	236814	64608	304

Many algorithms (such as SVM, K-Means) are sensitive to feature scale. A feature with a large range (e.g., 0-100,000) can "dominate" one with a small range (e.g., 0-1).

- **Objective**: Bring all features to a comparable scale.
- Improves the training speed (**convergence**) of the model.
- **Prevents** some **features** from **being weighted more than others** simply because of their range.
- *Essential for distance-based algorithms and neural networks.*

Normalization

Standardization

The scaling operation is applied per column.

Data Pre-processing – Feature Scaling

Normalization

Scales the data to a **fixed interval**, usually **[0, 1].**

Formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Pros: Useful when the **data distribution is unknown** or for algorithms such as neural networks.

Cons: Very **sensitive to outliers** (anomalous values).

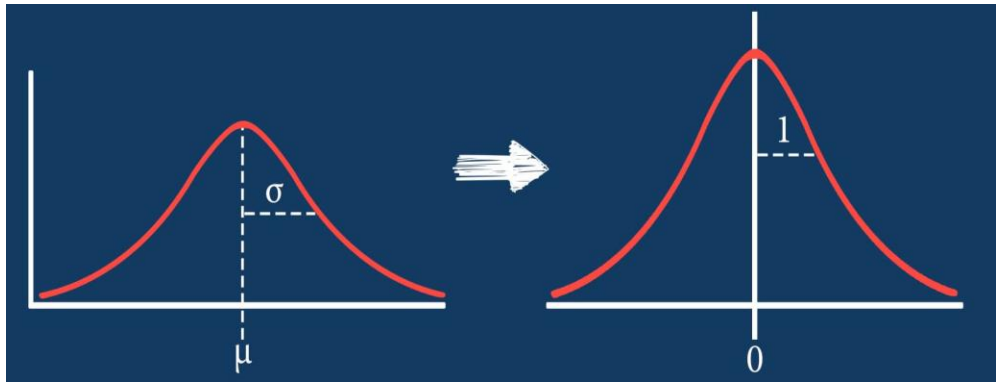
Data Pre-processing – Feature Scaling

Standardization

Transforms the data to have a **mean of 0** and a **standard deviation of 1**. So the interval is not fixed, **[-n, n].**

Formula:

$$X_{std} = \frac{X - \mu}{\sigma}$$

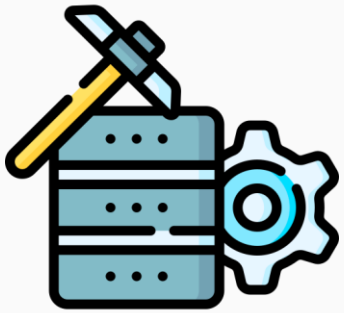


Pros: **Less sensitive to outliers**. It is the most common method for many ML algorithms.

Cons: Does **not constrain** values to a **fixed range.**

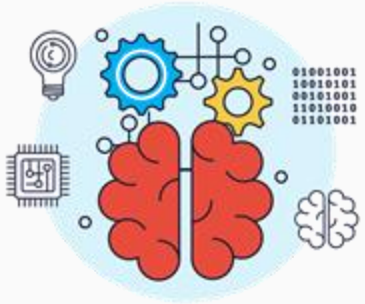
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling

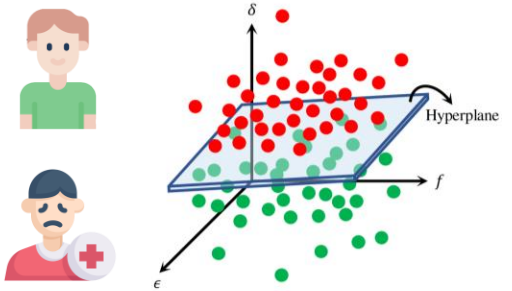
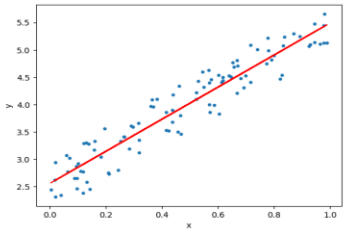
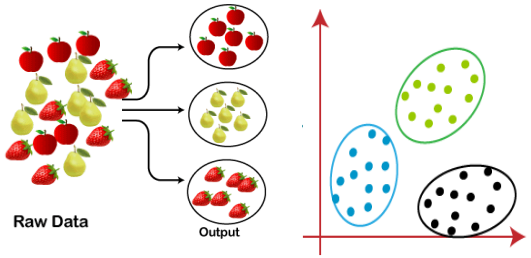


- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

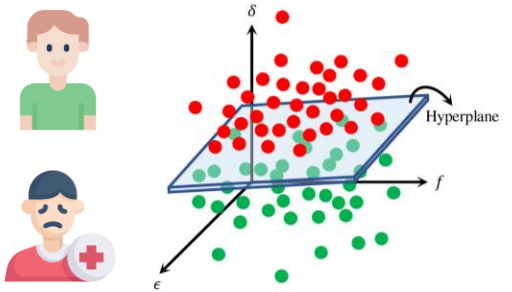
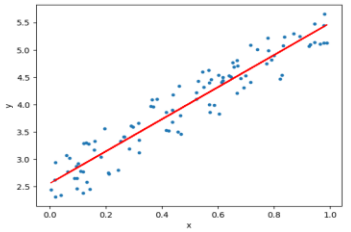
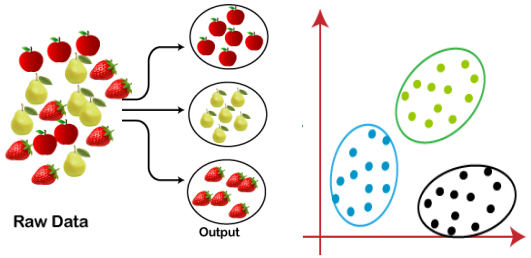
Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

Data Analysis - Tasks

Task	Description	Example
Classification	Assigning specific conditions or events to distinct categories based on signals, where data points are mapped to separate regions in feature space. - Finite classes	
Regression	Predict continuous values by modeling the underlying relationship or distribution of data in feature space. Example: Estimation of HRV (Heart Rate Variability) metrics from ECG signals. - Infinite outputs	
Clustering	Group data points into clusters by maximizing intra-cluster similarity and inter-cluster separation in the feature space. Example: Grouping ECG patterns to stratify patients based on cardiac profiles. - Finite groups	

Data Analysis - Tasks

Task	Description	Example
Classification	Assigning specific conditions or events to distinct categories based on signals, where data points are mapped to separate regions in feature space. - Finite classes	
Regression	Predict continuous values by modeling the underlying relationship or distribution of data in feature space. Example: Estimation of HRV (Heart Rate Variability) metrics from ECG signals. - Infinite outputs	
Clustering	Group data points into clusters by maximizing intra-cluster similarity and inter-cluster separation in the feature space. Example: Grouping ECG patterns to stratify patients based on cardiac profiles. - Finite groups	

Modelling - Classification

We have a dataset with cats and dogs and we want to build a model to classify them.

To do this, we need some **features**. Example features: height, weight, color, body shape etc.

Modelling - Classification

- **How do we separate these points?**

We have data on two dependent variables: **height** and **weight**.

How can we find a boundary between these two classes?

There are several possibilities

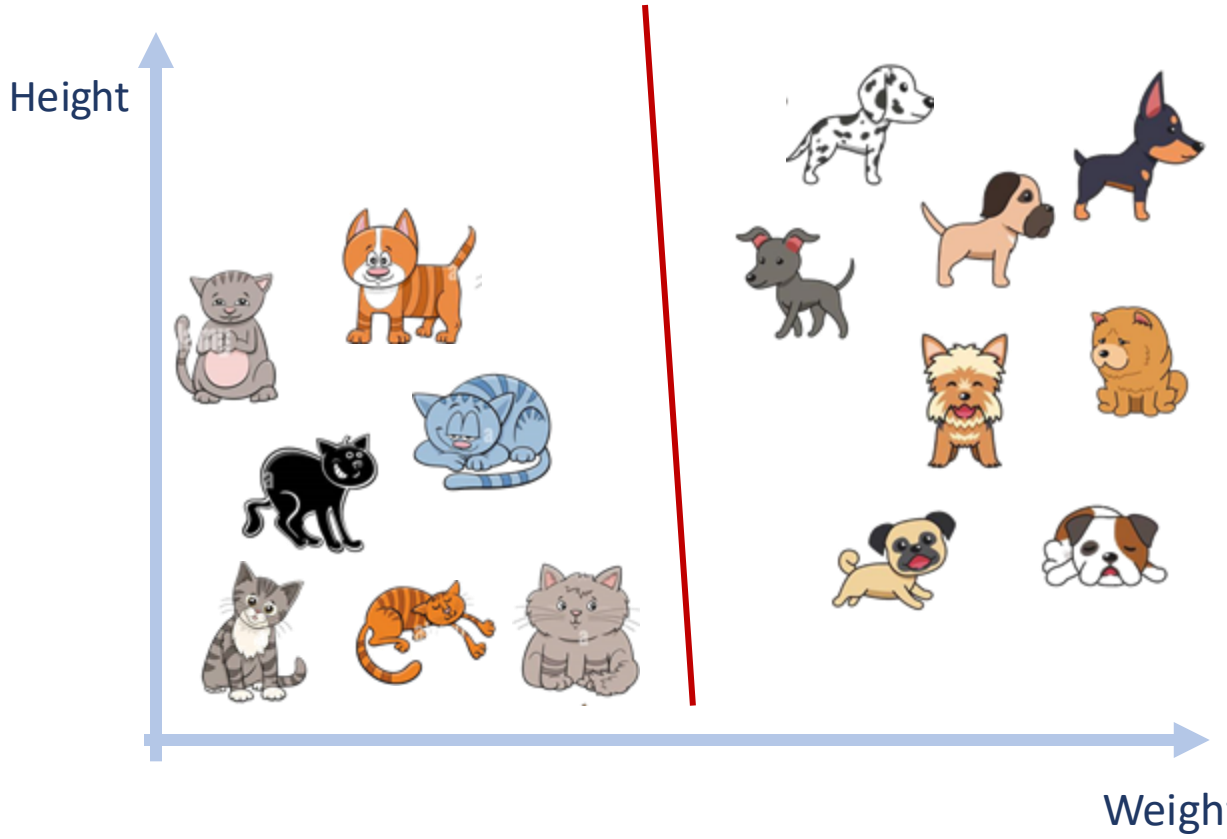
Modelling - Classification

- **How do we separate these points?**

We have data on two dependent variables: **height** and **weight**.

How can we find a boundary between these two classes?

There are several possibilities



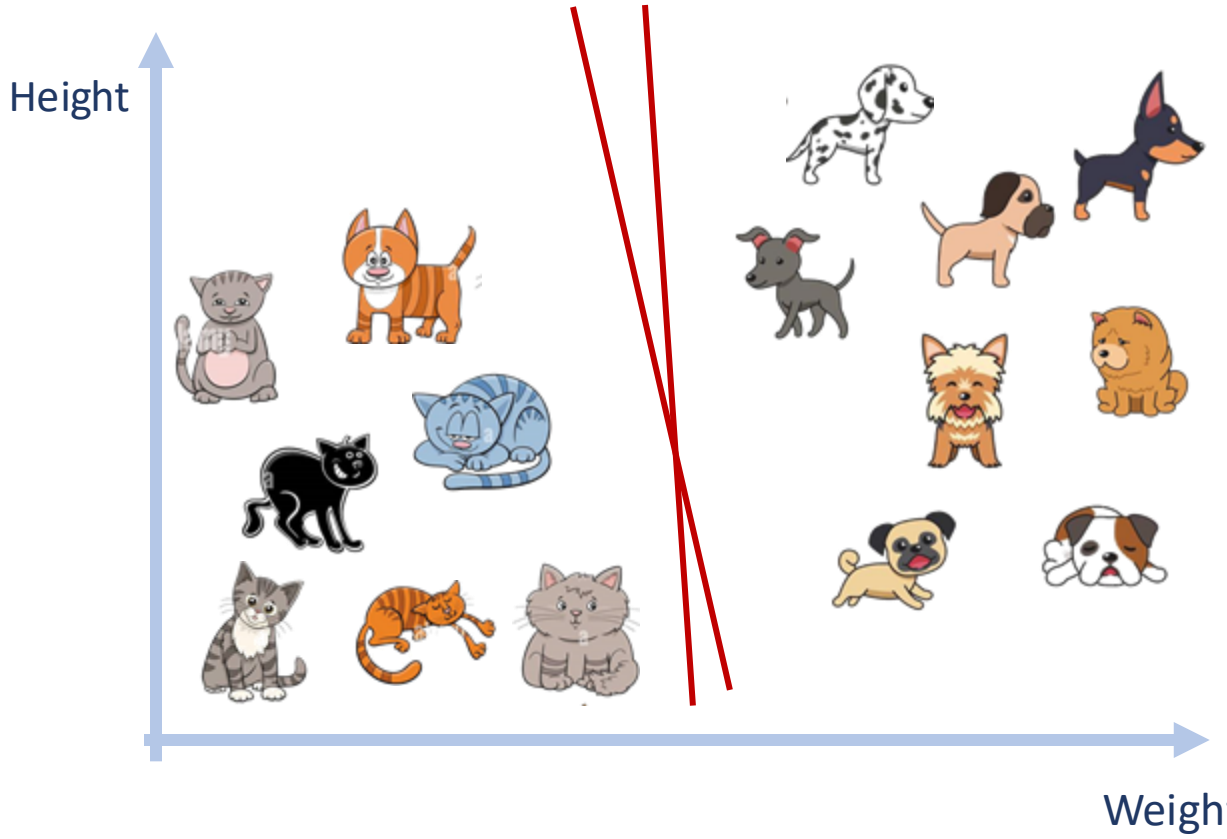
Modelling - Classification

- **How do we separate these points?**

We have data on two dependent variables: **height** and **weight**.

How can we find a boundary between these two classes?

There are several possibilities



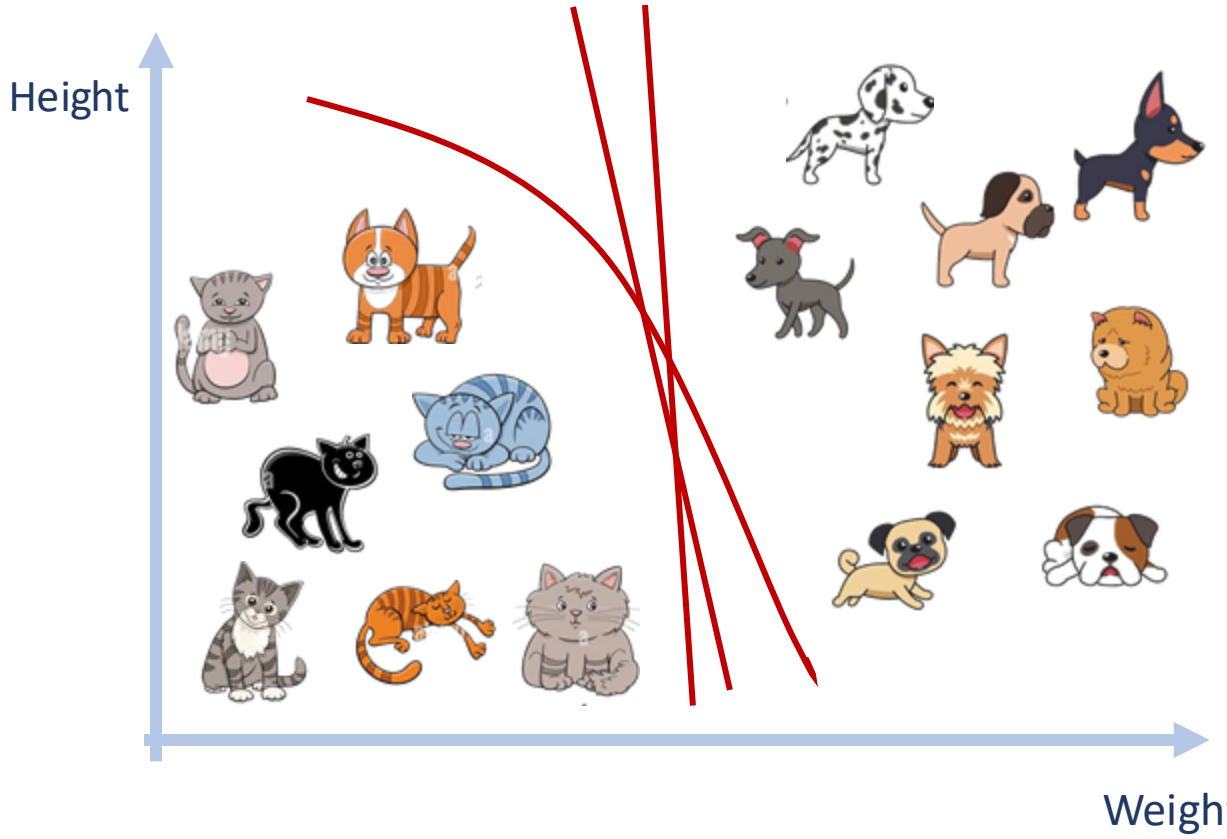
Modelling - Classification

- **How do we separate these points?**

We have data on two dependent variables: **height** and **weight**.

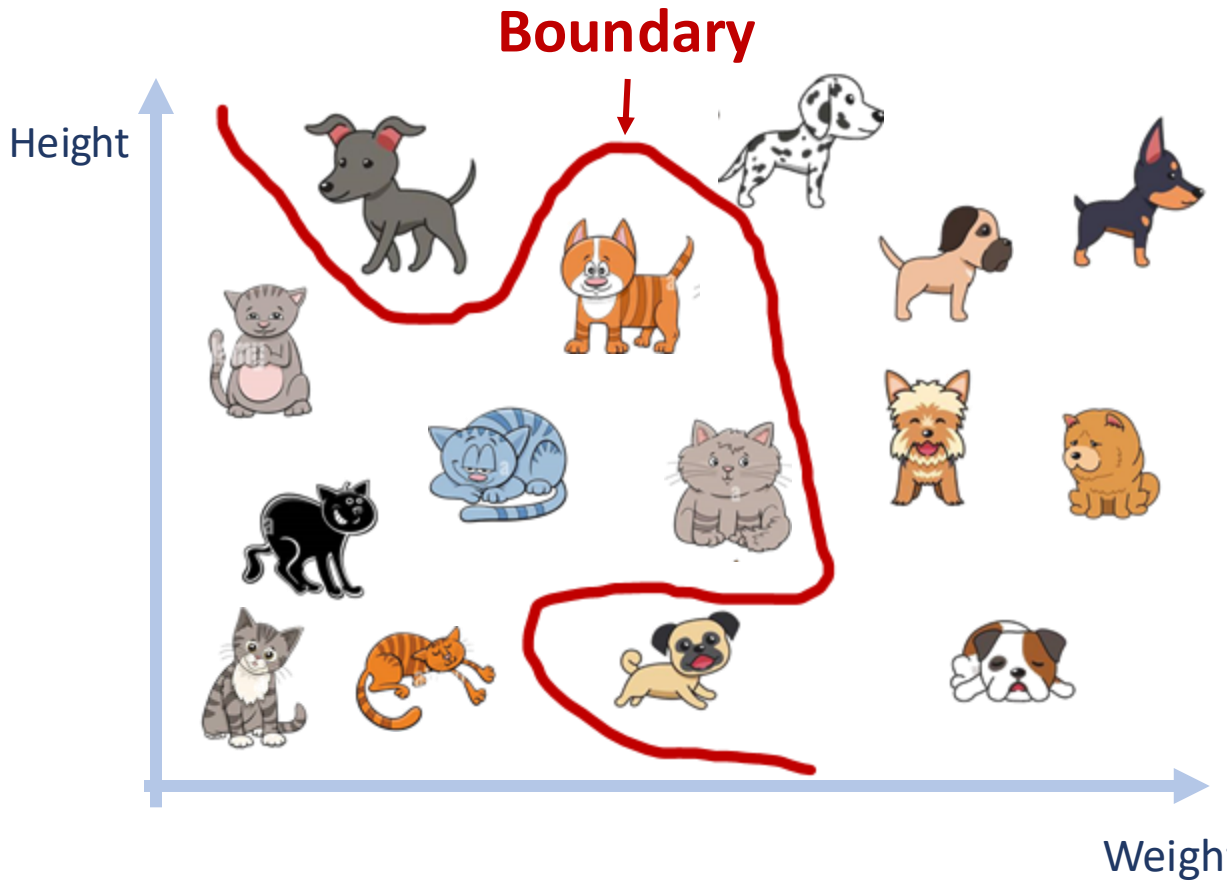
How can we find a boundary between these two classes?

There are several possibilities



Modelling – Classification

- Classes are **not well separated**
- The model creates a **very complex boundary**
- Perfect separation on **training data**



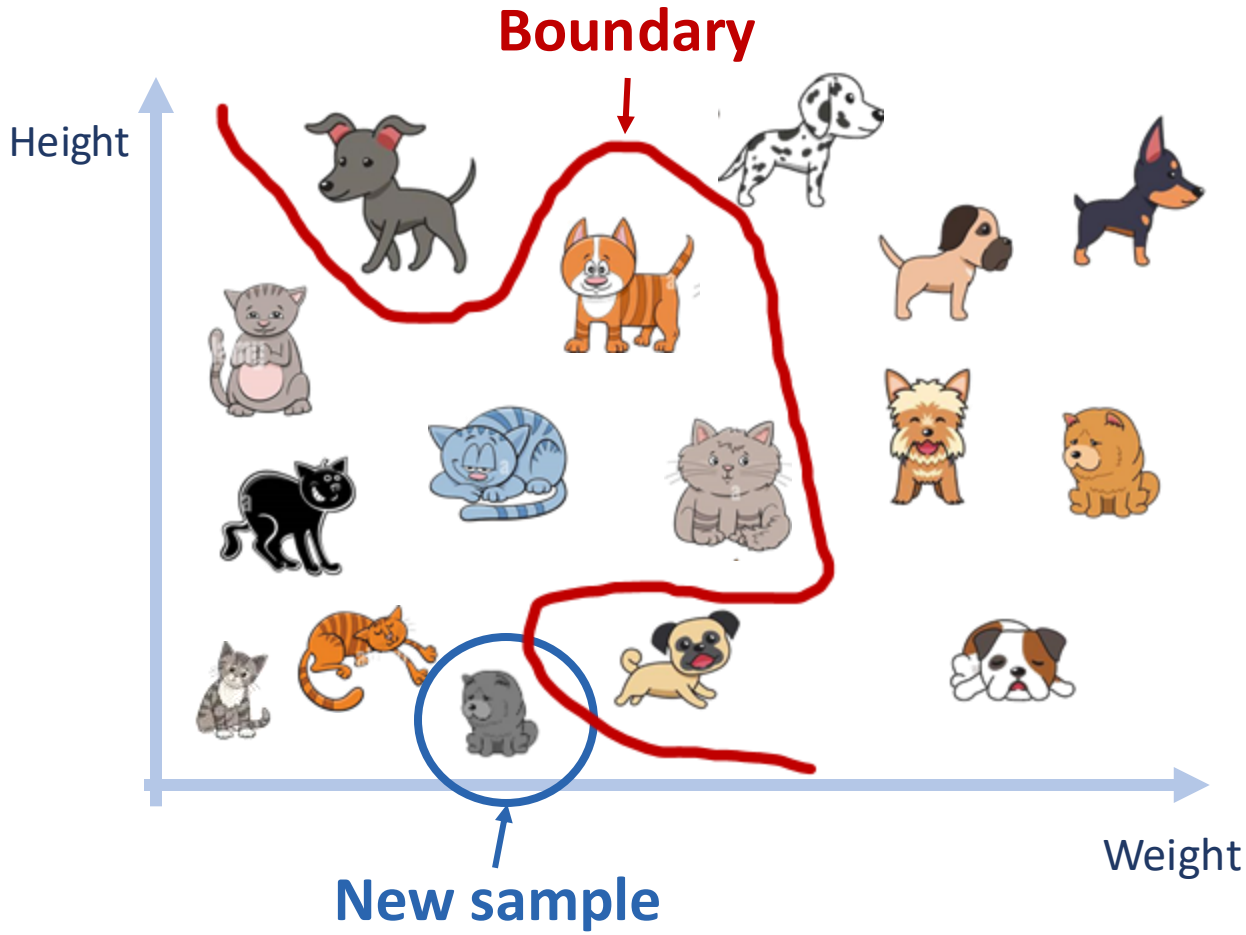
Modelling – Classification - Overfitting

If we now add a **new sample**, the model makes wrong predictions.

It has learned the training data "too well", it memorises details instead of the general pattern.

OVERFITTING

An overfitted model **cannot generalise**.

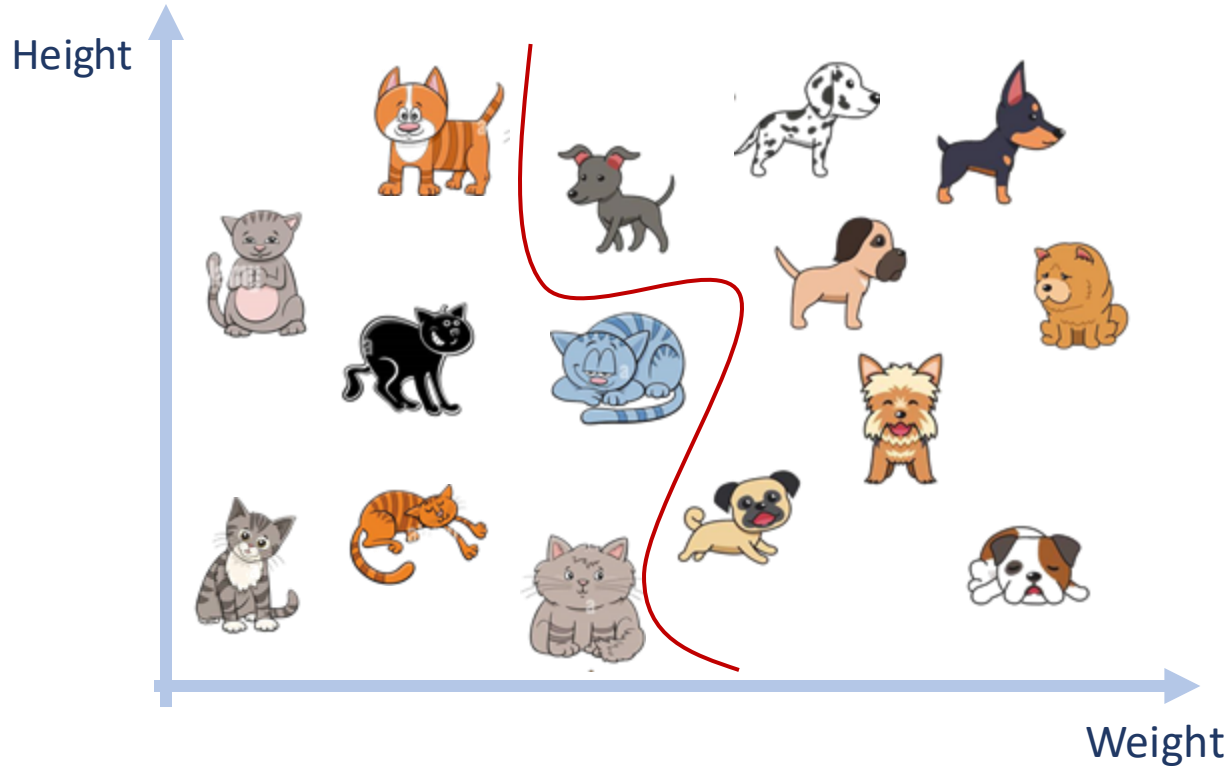


Modelling – Classification - Support Vector Machine (SVM)

In classification, many boundaries can separate the classes, but we want the **best boundary.**

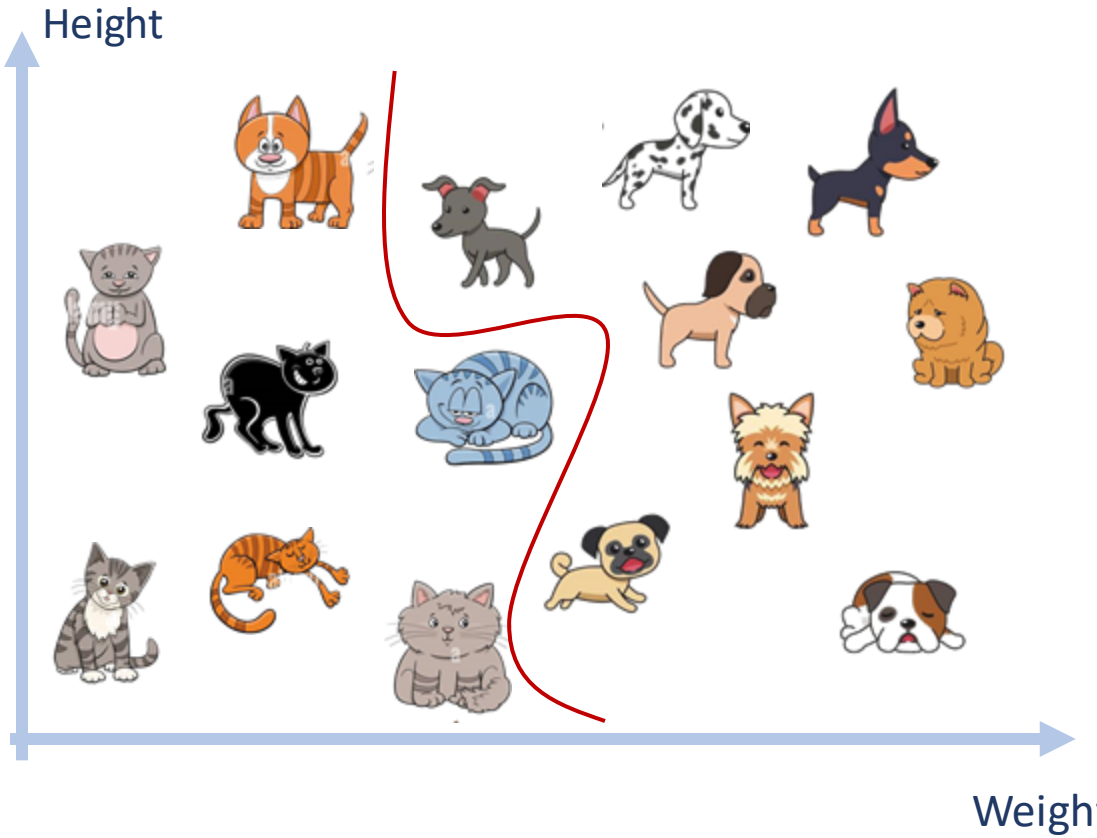
Which one should we choose?

A **Support Vector Machine** looks for the best boundary, the one that leaves the largest possible distance, **the margin**, between the boundary and the closest points of each class.



Modelling – Classification - Support Vector Machine (SVM)

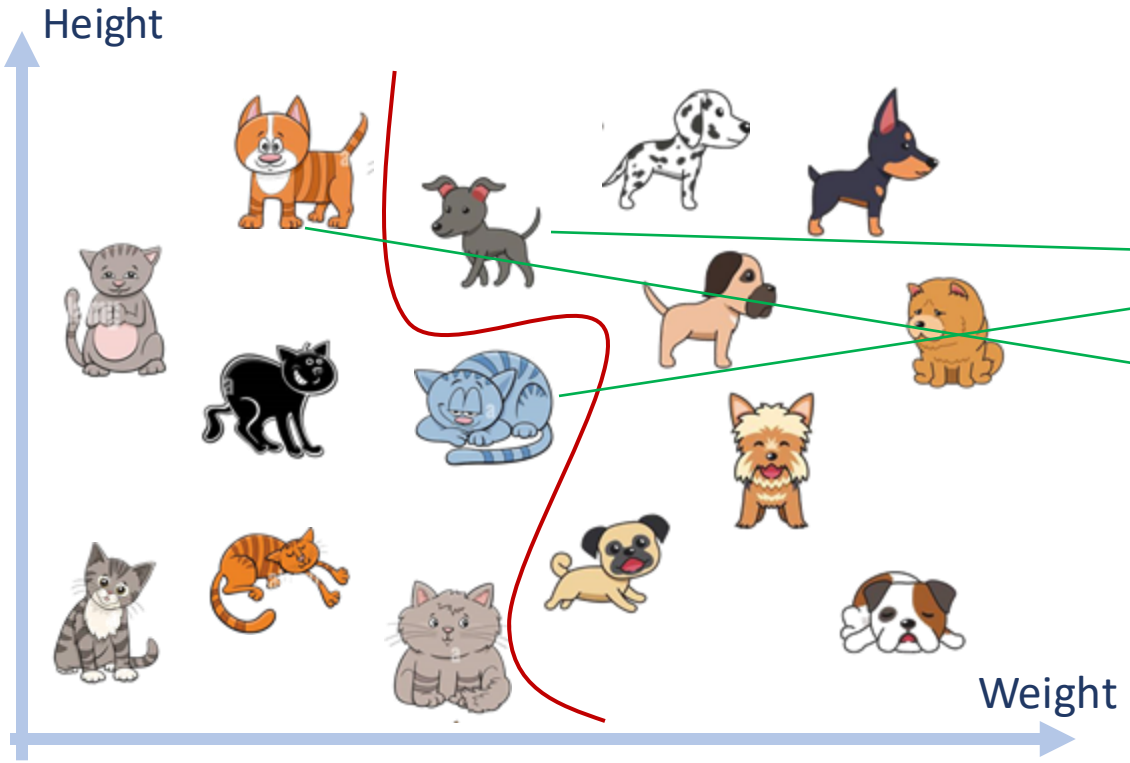
Input Space



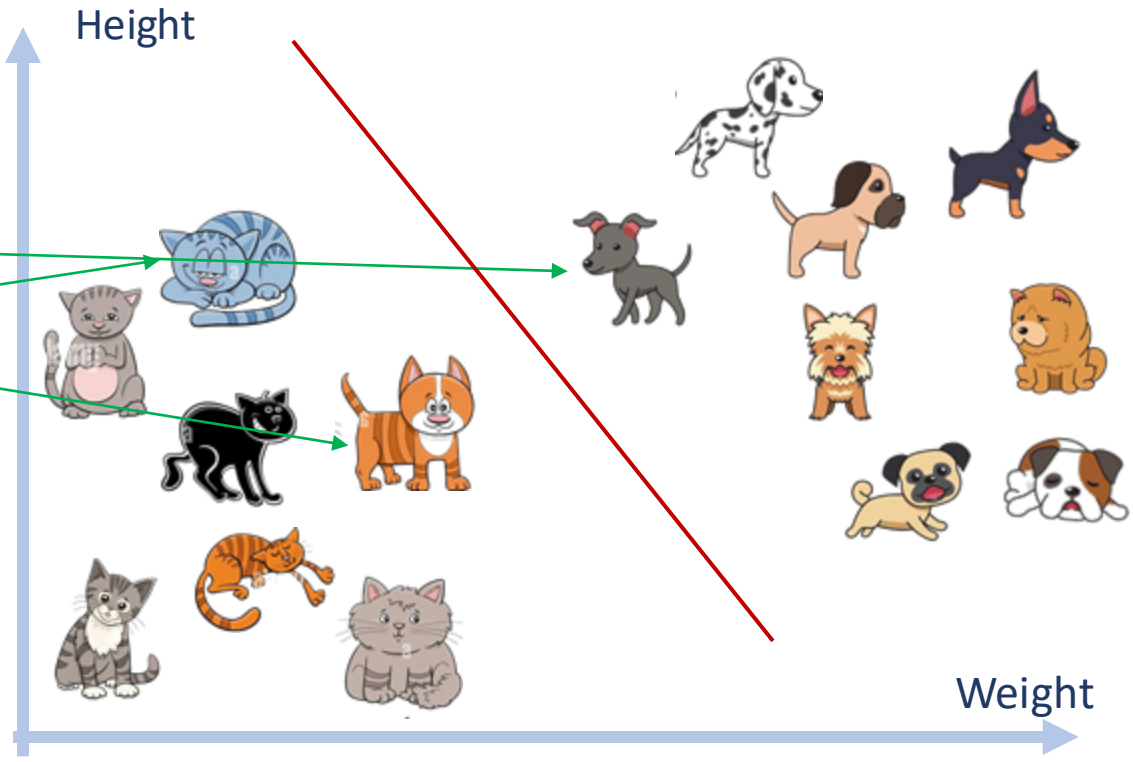
- Classes not well separated
- Complex boundary
- Overfitting

Modelling – Classification - Support Vector Machine (SVM)

Input Space



Features Space

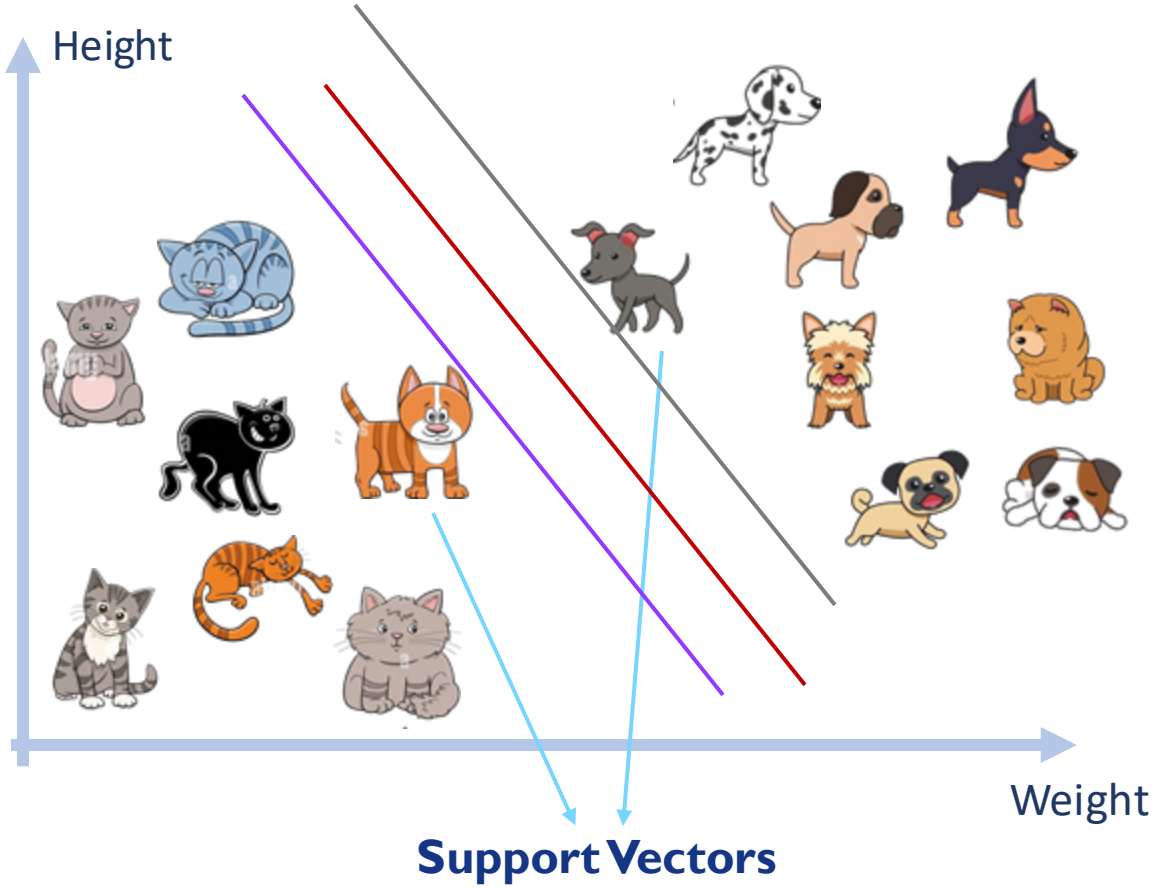


It chooses to use a mathematical transformation, called a **kernel**, to move the data into a new space where the classes become linearly separable.

Modelling – Classification - Support Vector Machine (SVM)

The SVM focuses only on the most 'difficult' points, **the ones closest** to the boundary.

In this case, looking at cats that look like dogs, and dogs that look more like cats.

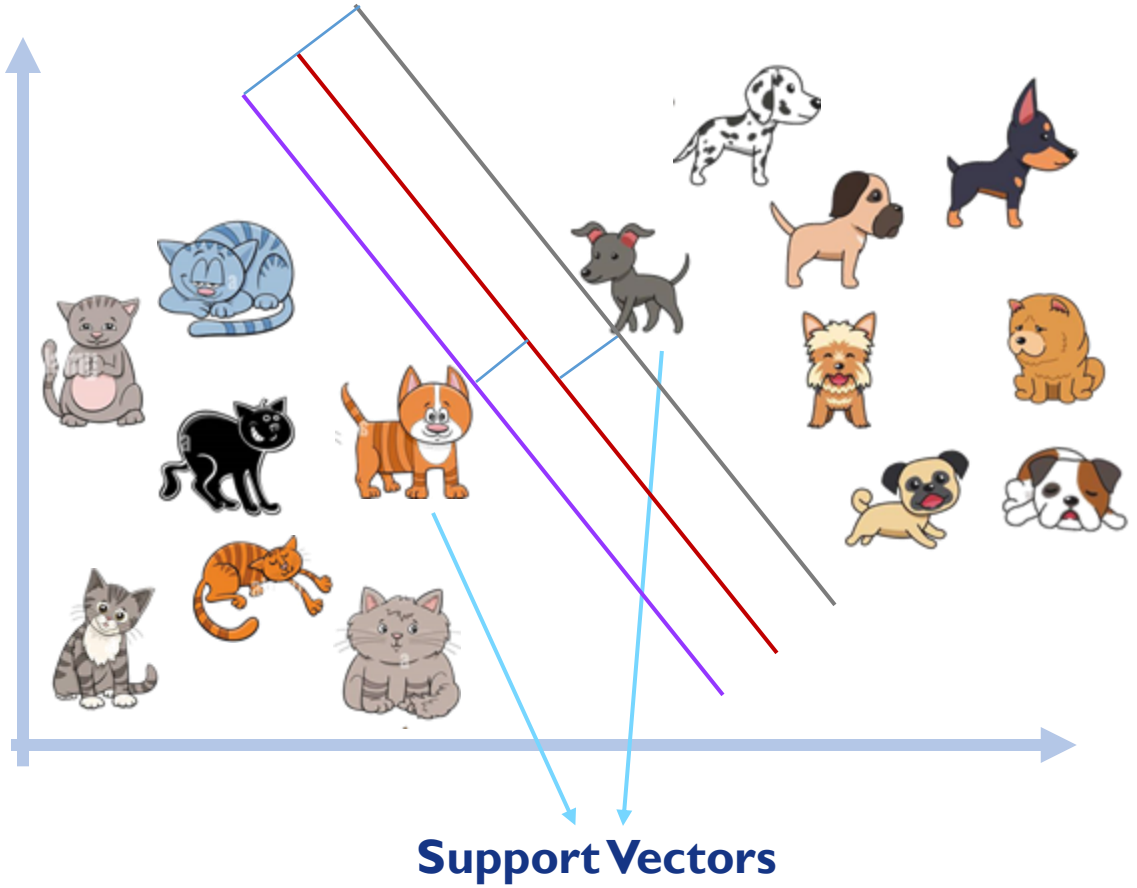


Modelling – Classification - Support Vector Machine (SVM)

The SVM focuses only on the most 'difficult' points, **the ones closest** to the boundary.

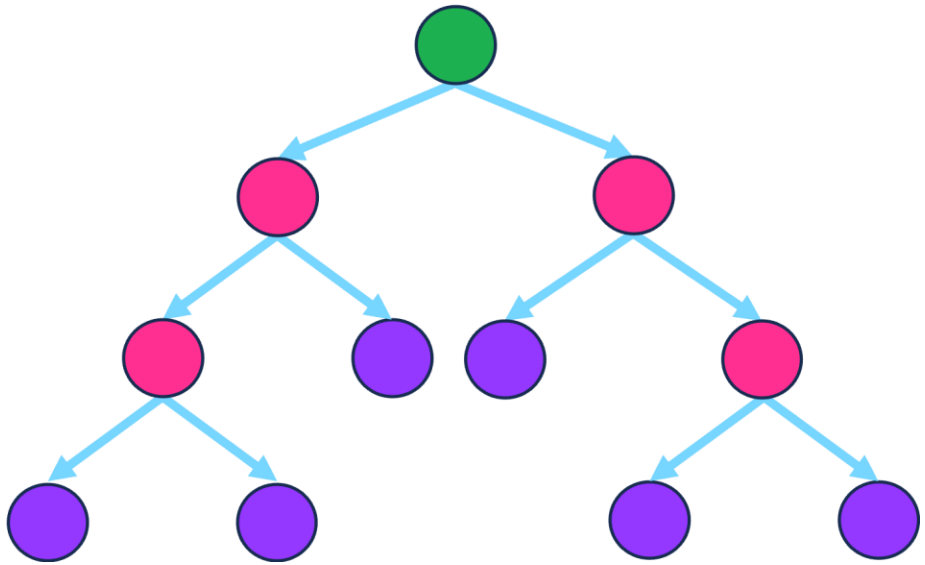
In this case, looking at cats that look like dogs, and dogs that look more like cats.

SVM searches for the boundary that **maximizes the minimum distance** (the margin) between the closest data points in each class (called **support vectors**) and the boundary.



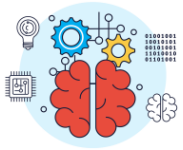
Modelling – Classification – Decision Tree

It predicts the target label by **learning simple decision rules** deduced from data characteristics.



It looks like a flowchart, where:

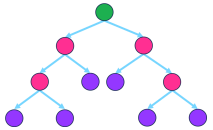
- **Root node**: The initial node at the beginning of a decision tree, where the entire population or data set begins to divide based on various features or conditions.
- **Branches**: Divide the data set based on feature values. Represents a specific decision path.
- **Internal nodes**: Represent the conditions (decisions) based on the features.
- **Leaf nodes**: Nodes for which further subdivision is not possible; they represent the class labels (outcomes).



Modelling – Classification – Decision Tree

Example: Predicting Heart Attack Risk with a Decision Tree

1. **Features:** Age, Weight, Smoking status.
2. **Target:** "High Risk" or "Low Risk"

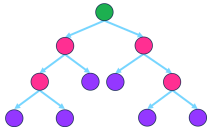


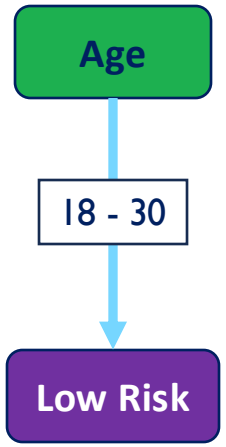
Age

Modelling – Classification – Decision Tree

Example: Predicting Heart Attack Risk with a Decision Tree

1. **Features:** Age, Weight, Smoking status.
2. **Target:** "High Risk" or "Low Risk"



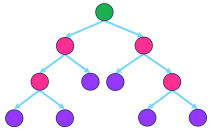


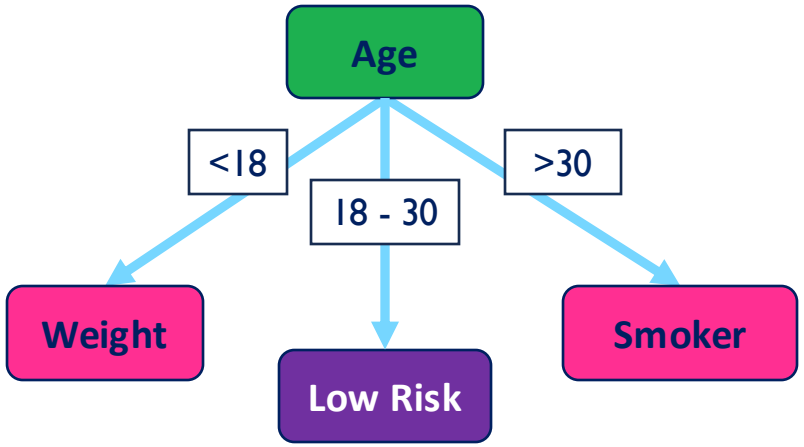
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Modelling – Classification – Decision Tree

Example: Predicting Heart Attack Risk with a Decision Tree

1. **Features:** Age, Weight, Smoking status.
2. **Target:** "High Risk" or "Low Risk"

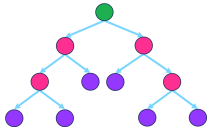


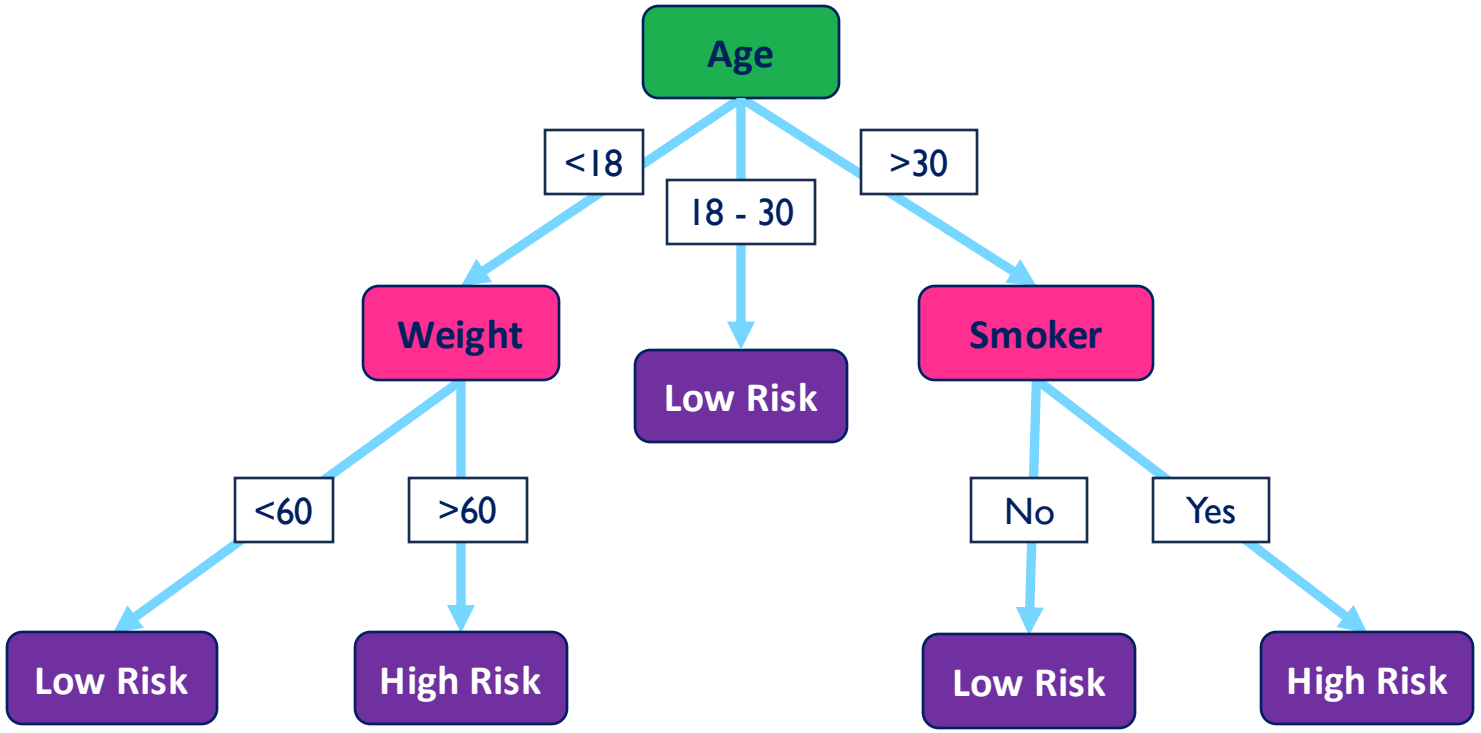


Modelling – Classification – Decision Tree

Example: Predicting Heart Attack Risk with a Decision Tree

1. **Features:** Age, Weight, Smoking status.
2. **Target:** "High Risk" or "Low Risk"



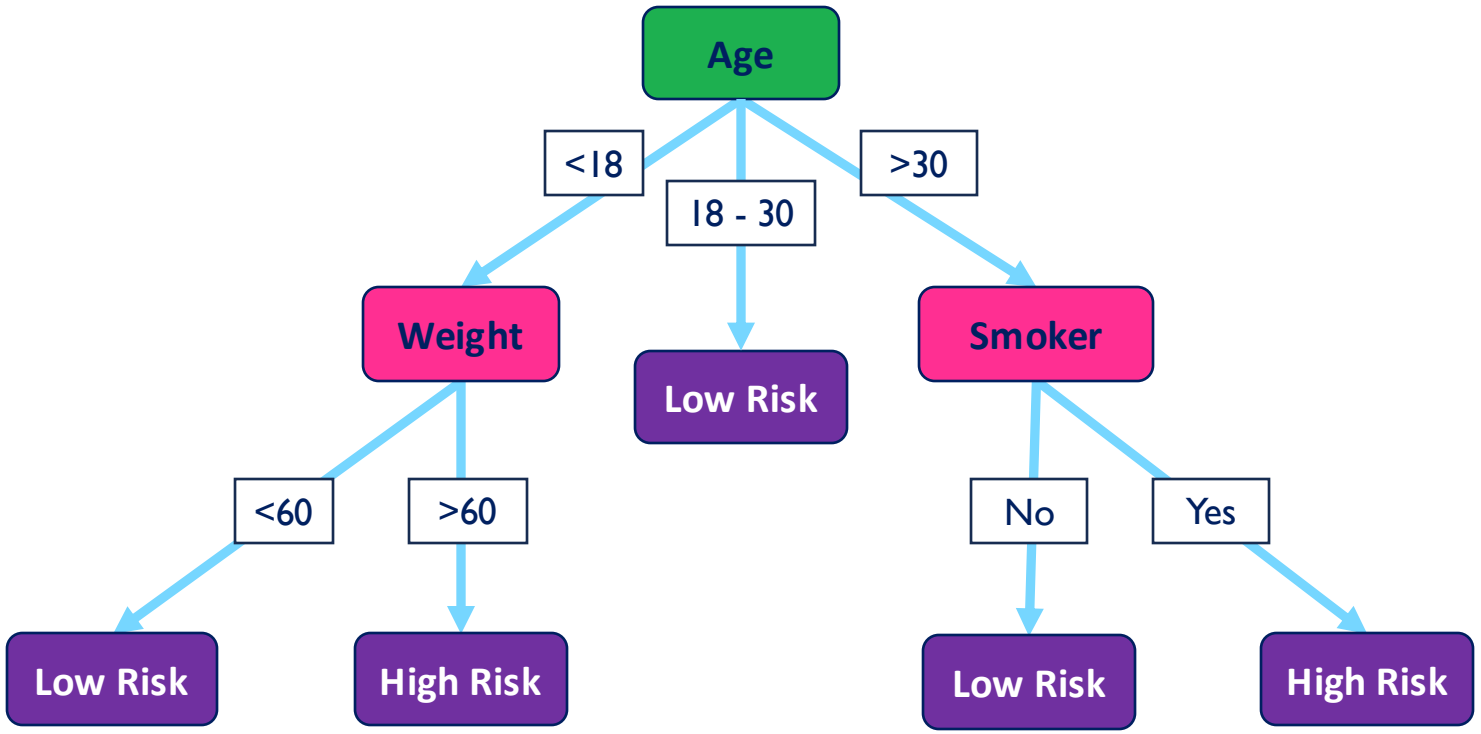


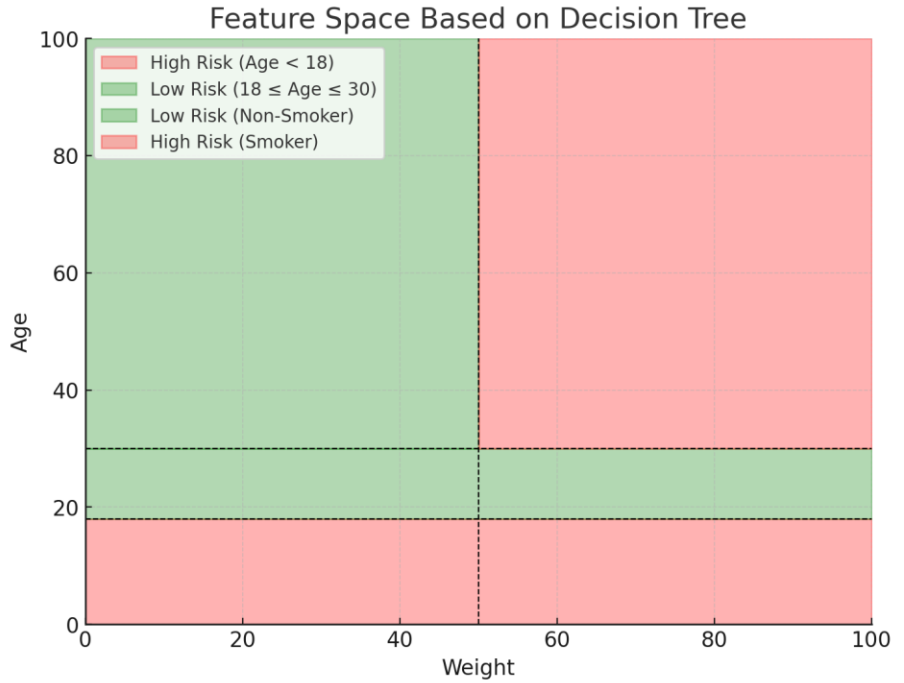
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Modelling – Classification – Decision Tree

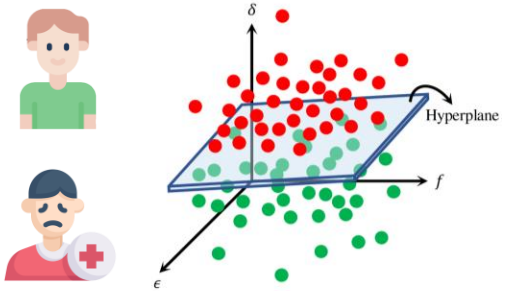
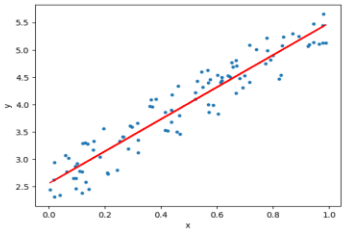
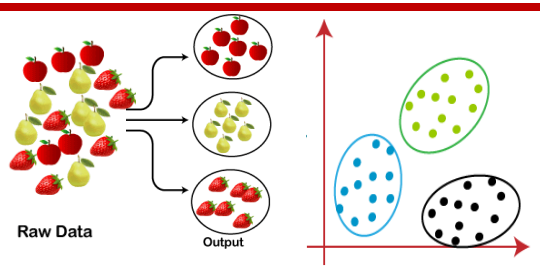
Example: Predicting Heart Attack Risk with a Decision Tree

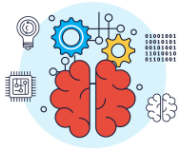
1. **Features:** Age, Weight, Smoking status.
2. **Target:** "High Risk" or "Low Risk"





Data Analysis - Tasks

Task	Description	Example
Classification	Assigning specific conditions or events to distinct categories based on signals, where data points are mapped to separate regions in feature space. - Finite classes	
Regression	Predict continuous values by modeling the underlying relationship or distribution of data in feature space. Example: Estimation of HRV (Heart Rate Variability) metrics from ECG signals. - Infinite outputs	
Clustering	Group data points into clusters by maximizing intra-cluster similarity and inter-cluster separation in the feature space. Example: Grouping ECG patterns to stratify patients based on cardiac profiles. - Finite groups	



Modelling – Linear Regression Model

$$\hat{y} = b_0 + b_1 X_1$$

Dependent variable
(predicted value)

Intercept
(costant)

Slope coefficient

Independent variable
(input feature)

Modelling – Regression – Simple Linear Regression

Forecast the amount of potatoes on a farm harvest based on the amount of fertilizer the farmer use

~

Modelling – Regression – Simple Linear Regression

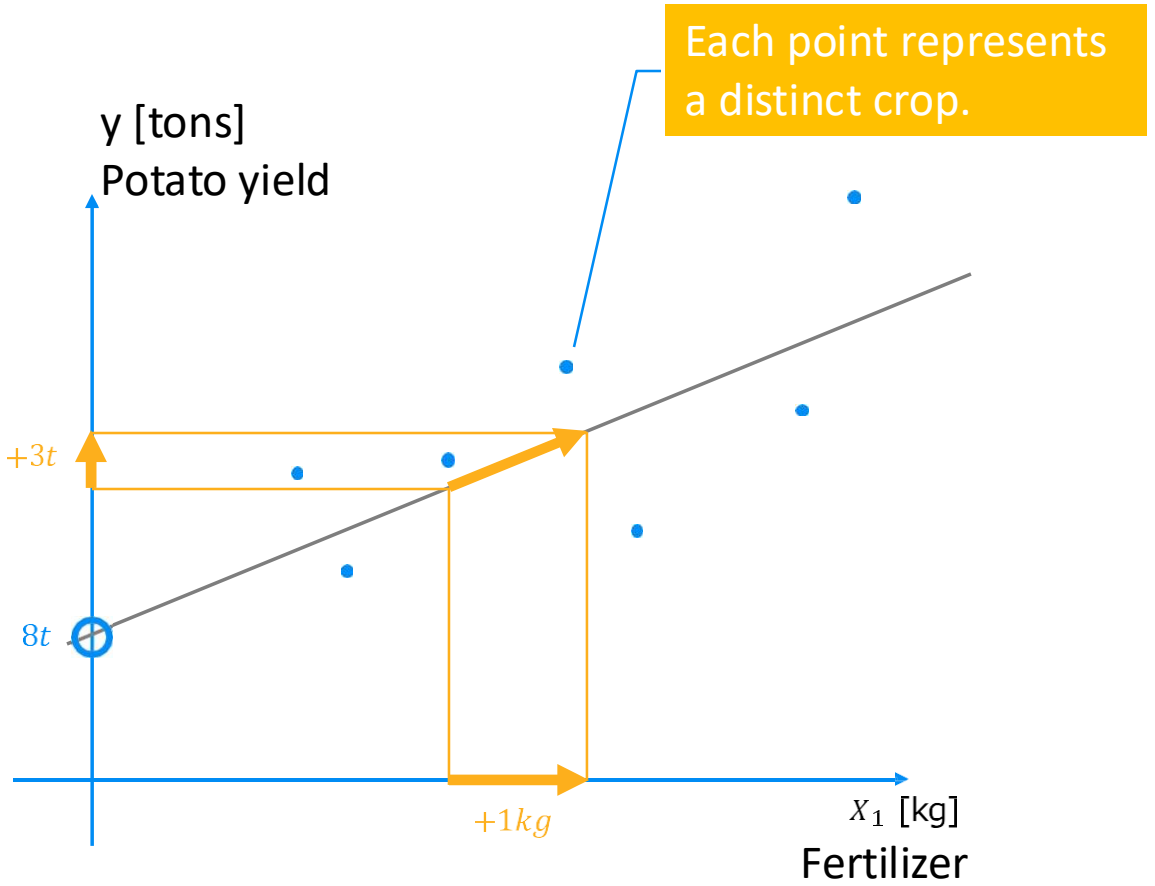
Forecast the amount of potatoes on a farm harvest based on the amount of fertilizer the farmer use

$$\text{Potatoes [t]} =$$
$$= b_0 + b_1 \times Fertilizer[kg]\hat{y} =$$
$$= b_0 + b_1 X_1$$

If we consider a linear regression we obtain:

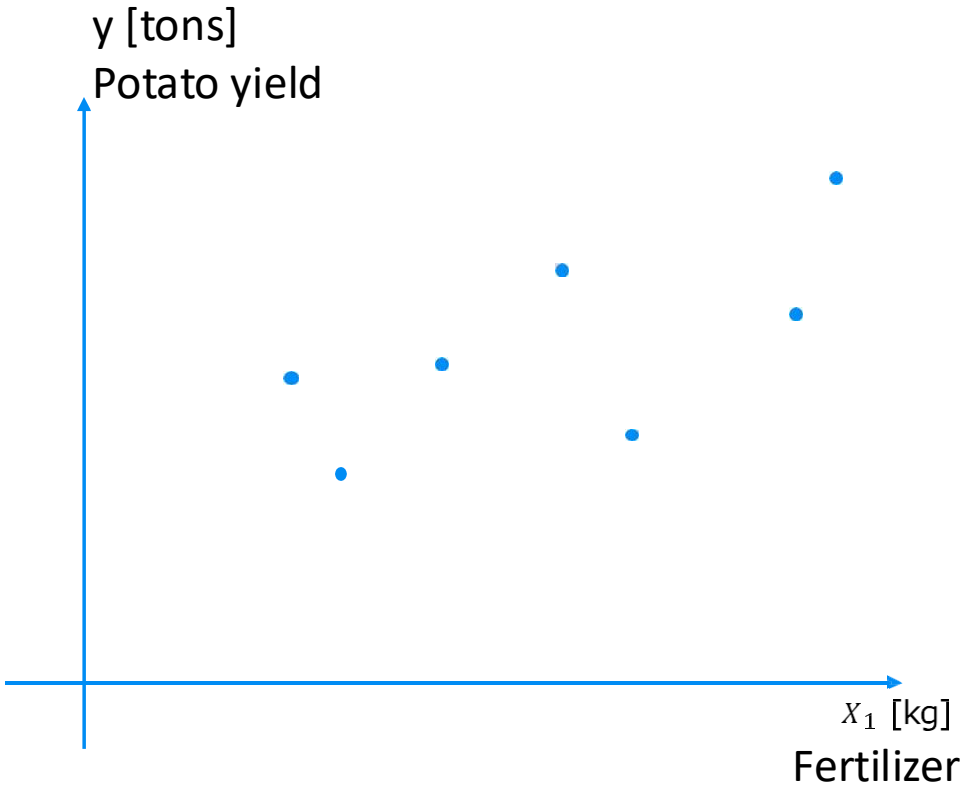
$b_0 = 8[t]$ Intercept

$b_1 = 3\ [\frac{t}{kg}]$ If X_1 increases by 1, how much does y increase?



Modelling – Regression – Simple Linear Regression


We collect crop data and graph it



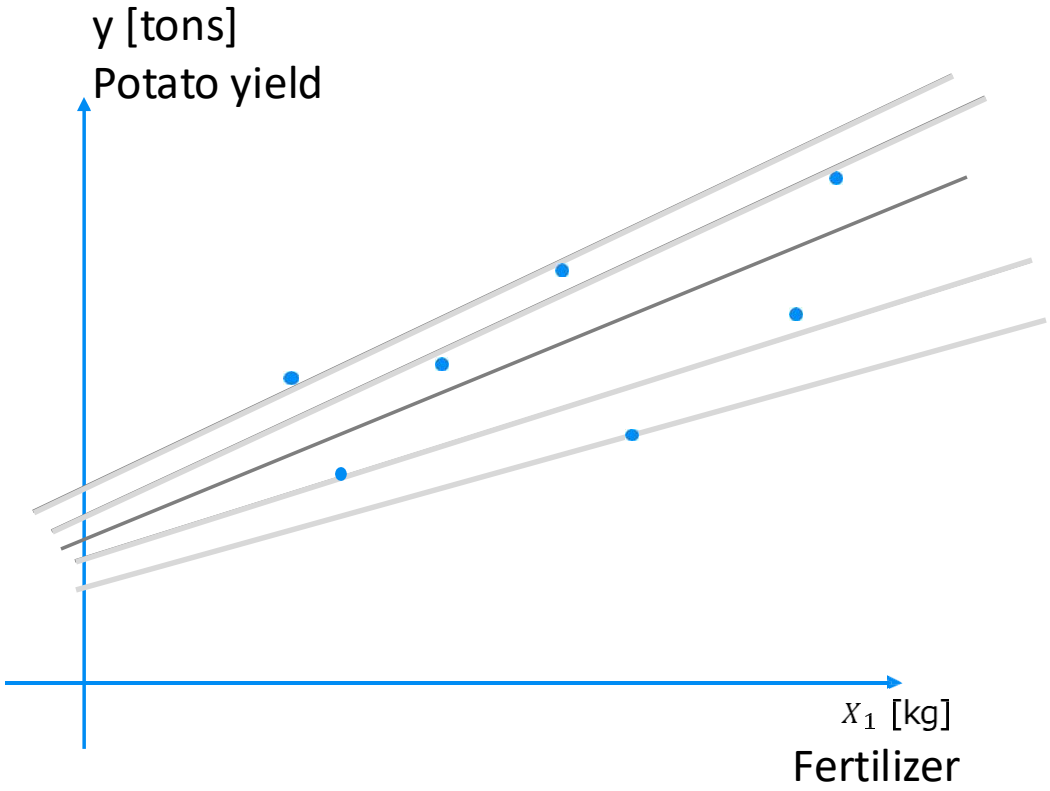
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Modelling – Regression – Simple Linear Regression

~

We collect crop data and graph it

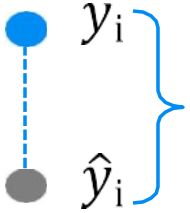
From the given points, which of the slopes is best?



Modelling – Regression – Simple Linear Regression

From the given points, which of the slopes is best? **OLS – Ordinary Least Squares** is the answer

We take our data points and **project them vertically** onto our linear regression line.
This operation must be performed for **all linear regression lines** under consideration.



y_i y_i it is the 'true' value

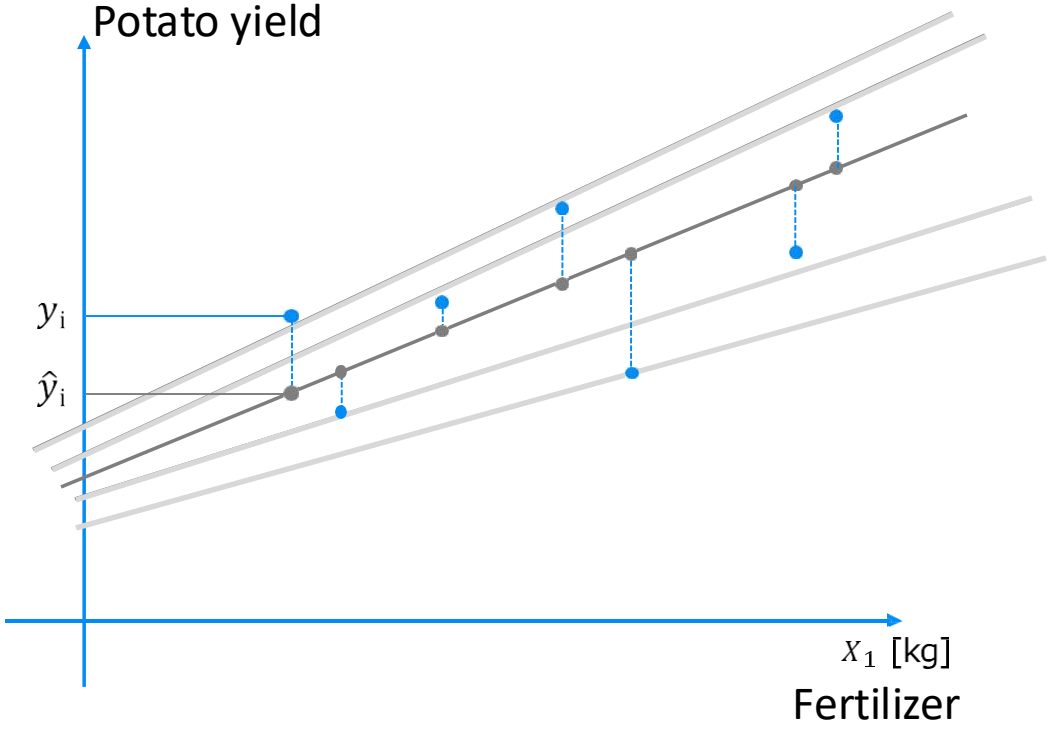
residual $\varepsilon_i = y_i - \hat{y}_i$

$\hat{y}_i$ $\hat{y}_i$ it is the predicted value

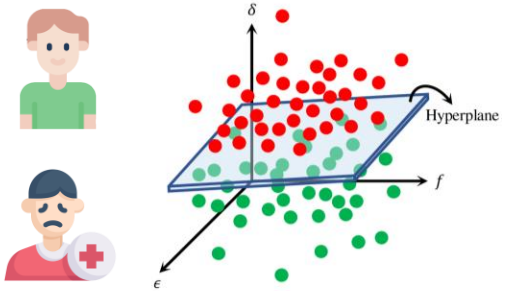
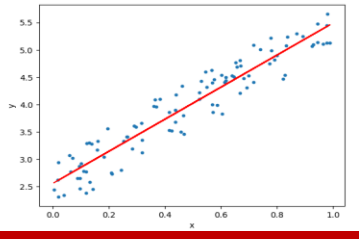
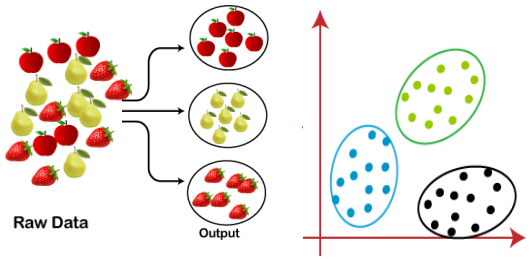
$$\hat{y} = b_0 + b_1 X_1$$

b_0, b_1 So that

$\sum (y_i - \hat{y}_i)^2$ is minimized

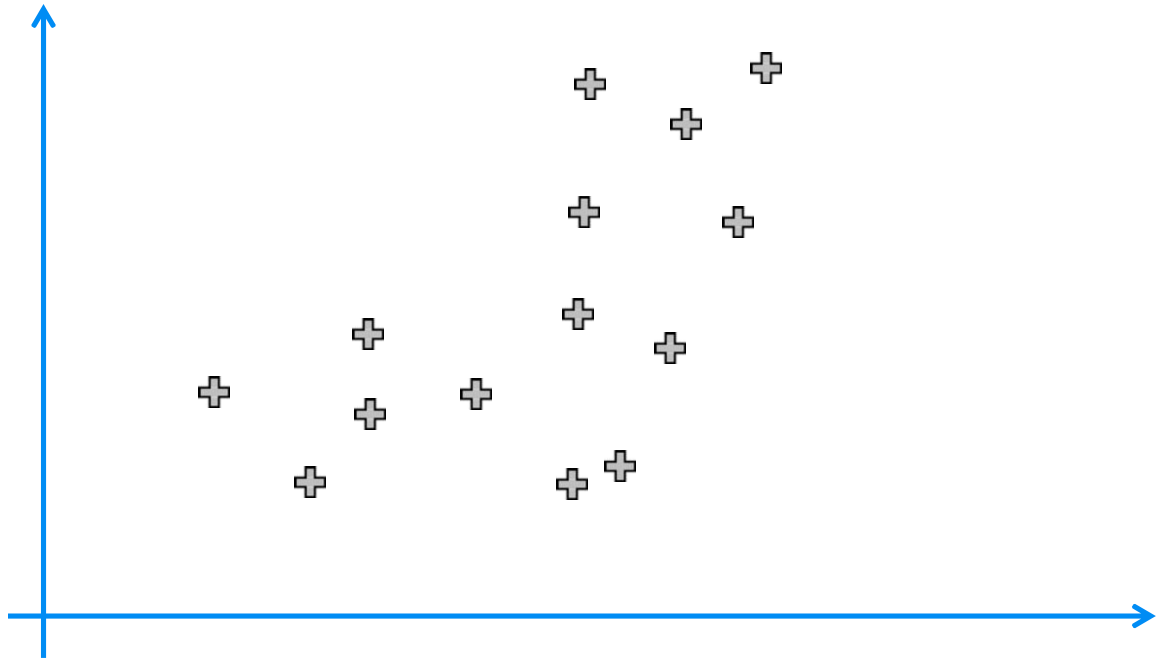


Data Analysis - Tasks

Task	Description	Example
Classification	Assigning specific conditions or events to distinct categories based on signals, where data points are mapped to separate regions in feature space. - Finite classes	
Regression	Predict continuous values by modeling the underlying relationship or distribution of data in feature space. Example: Estimation of HRV (Heart Rate Variability) metrics from ECG signals. - Infinite outputs	
Clustering	Group data points into clusters by maximizing intra-cluster similarity and inter-cluster separation in the feature space. Example: Grouping ECG patterns to stratify patients based on cardiac profiles. - Finite groups	

Modelling – Clustering – K-Means

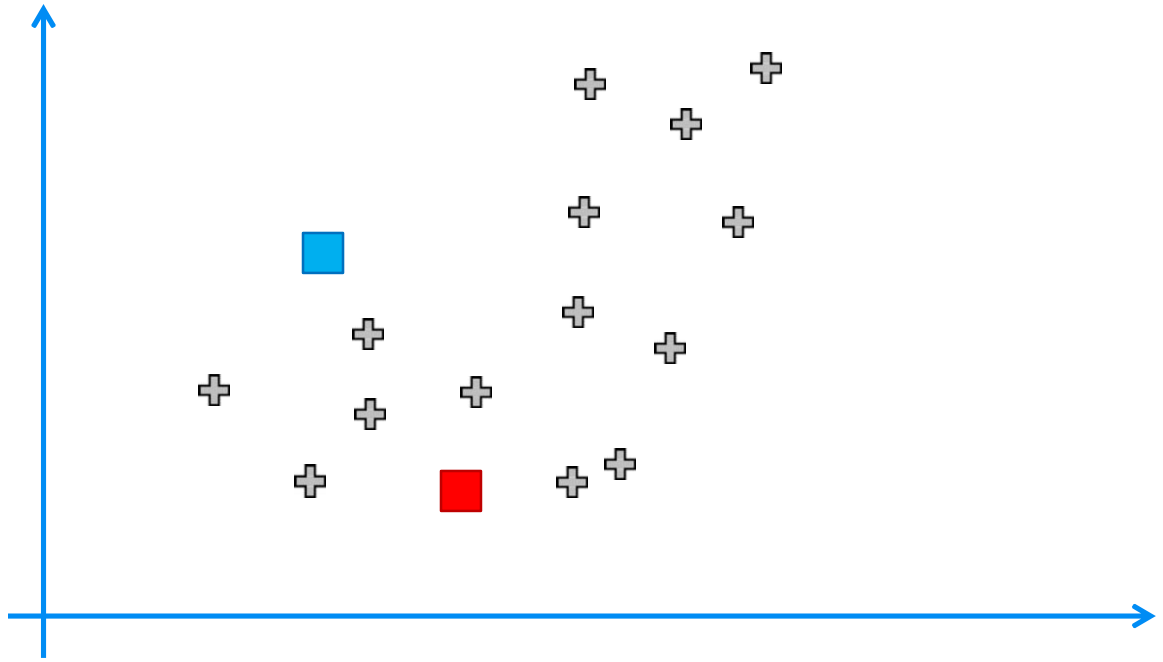
From the given points we want to create different groups, or better **clusters**



Modelling – Clustering – K-Means

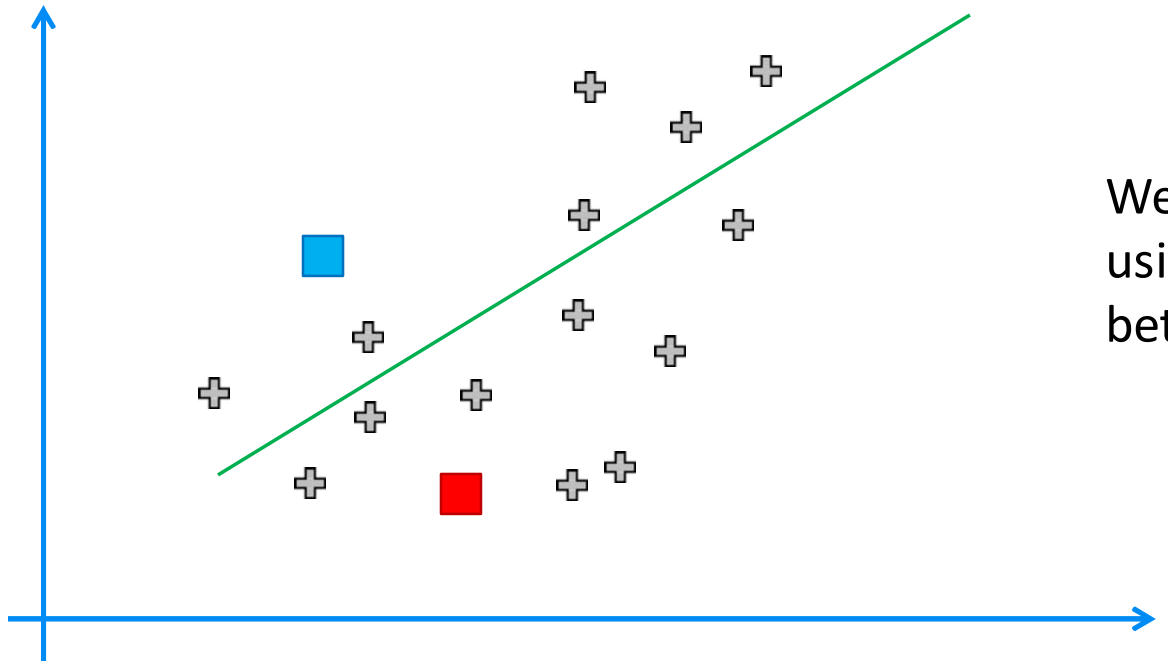
We have to decide how many clusters we want to form (e.g. 2).

For each cluster we randomly place a **centroid**



Modelling – Clustering – K-Means

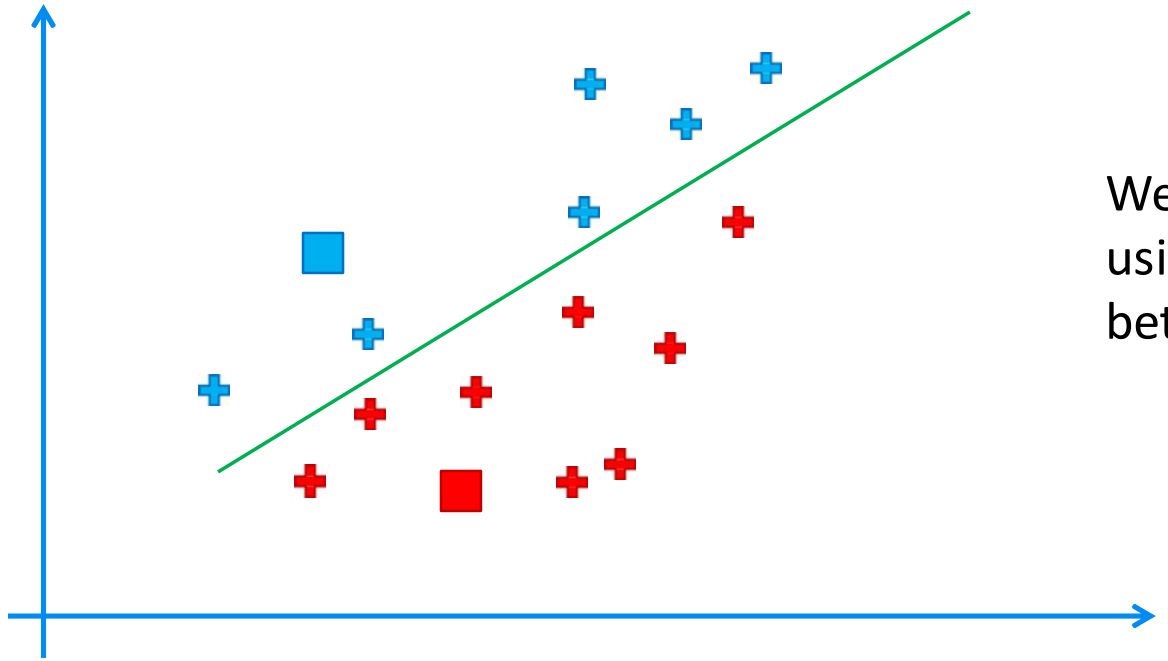
The K-Means method assigns each point to the **closest centroid**.



We can separate the points using an **equidistance line** between the two centroids

Modelling – Clustering – K-Means

The K-Means method assigns each point to the **closest centroid**.

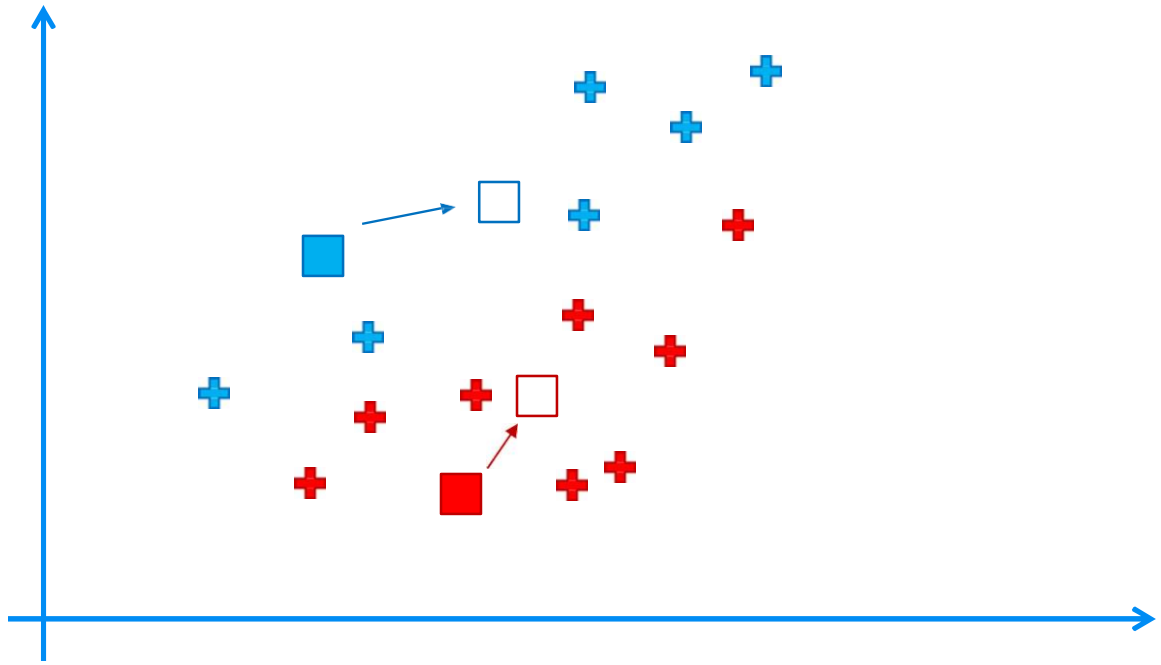


We can separate the points using an **equidistance line** between the two centroids

All points were associated to the cluster

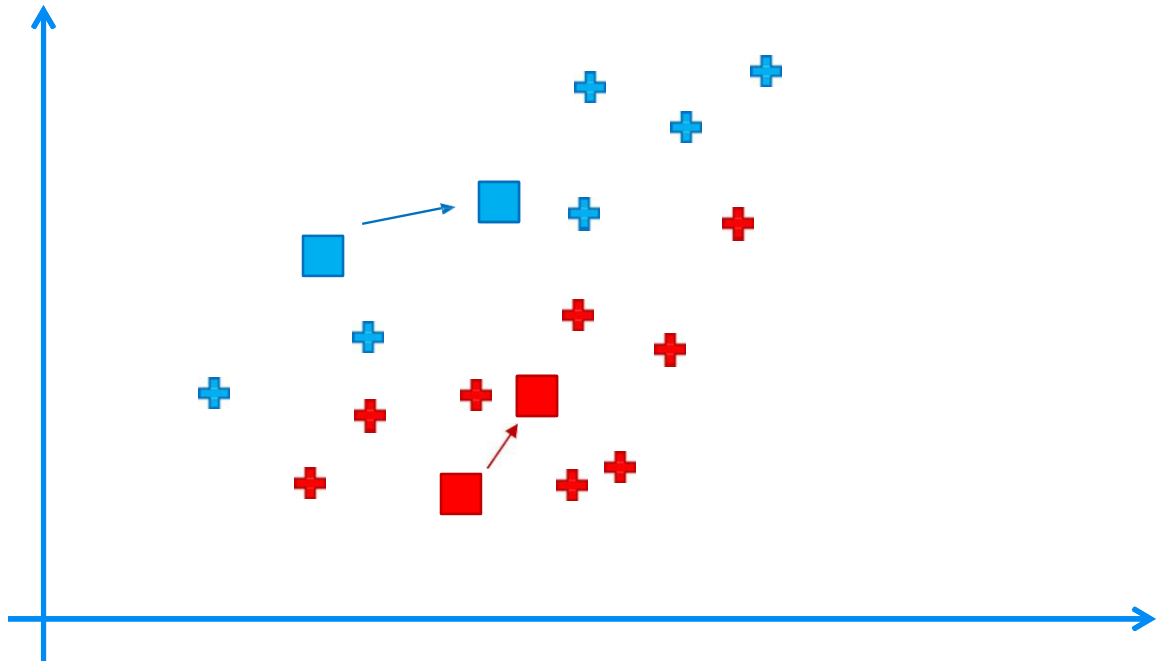
Modelling – Clustering – K-Means

For each cluster, we calculate a new centroid as the average of all assigned points.



Modelling – Clustering – K-Means

For each cluster, we calculate a new centroid as the average of all assigned points.

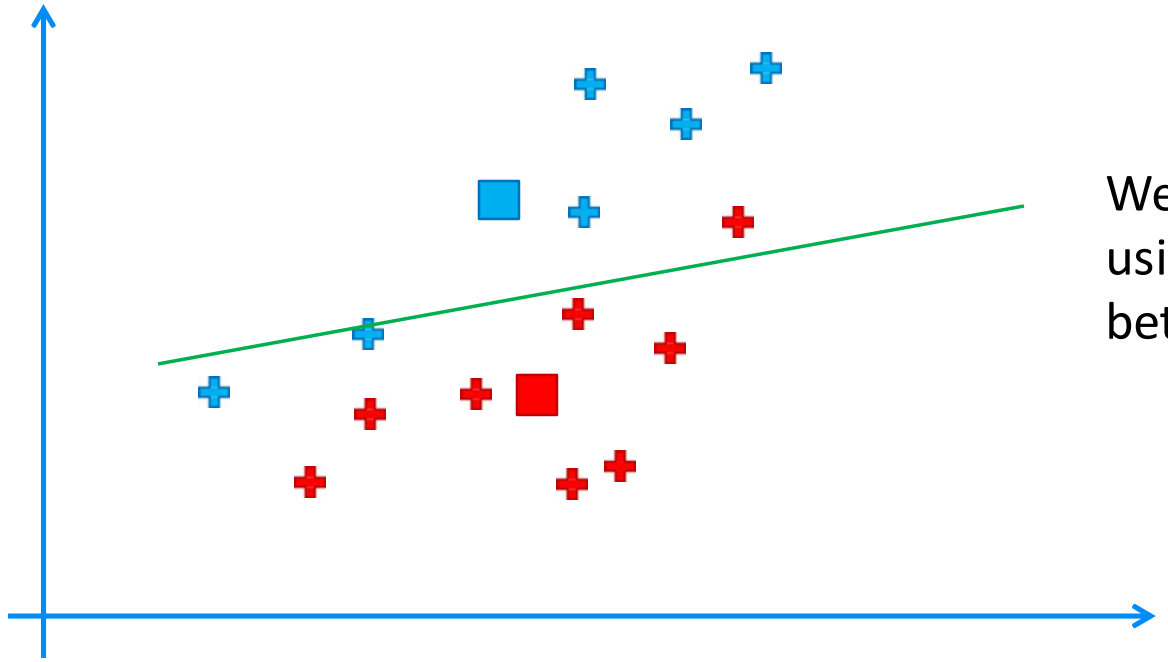


The new centroid has been calculated

Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

- Reassign
- Calculate the centroid
- Move the centroid.

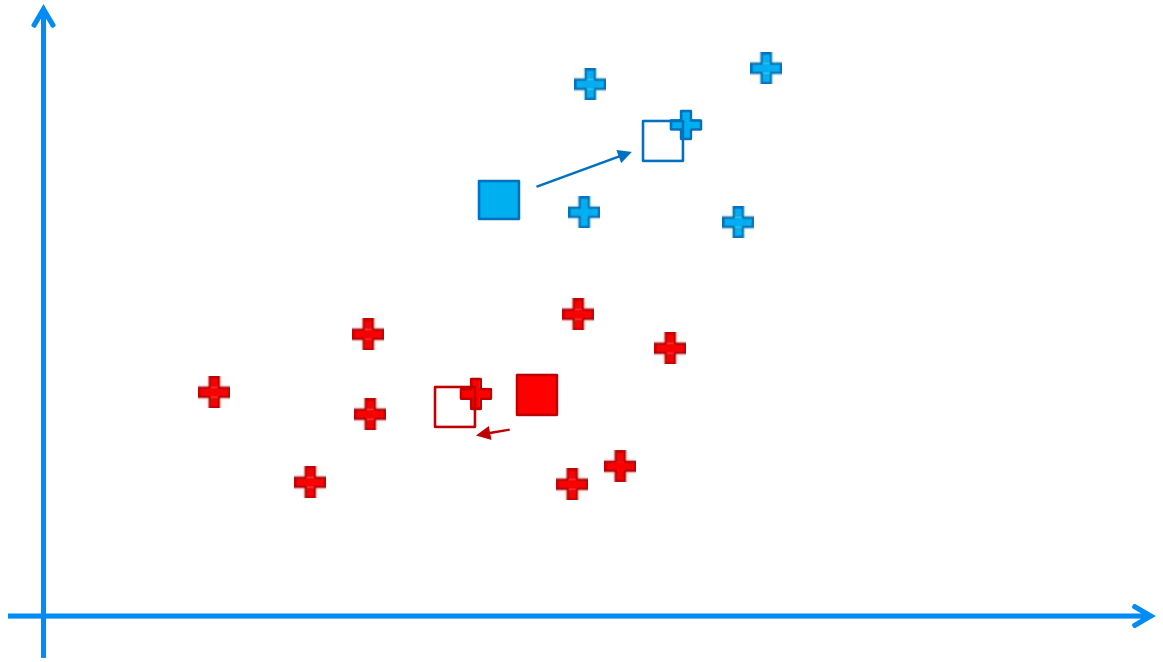


We can separate the points using an **equidistance line** between the two centroids

Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

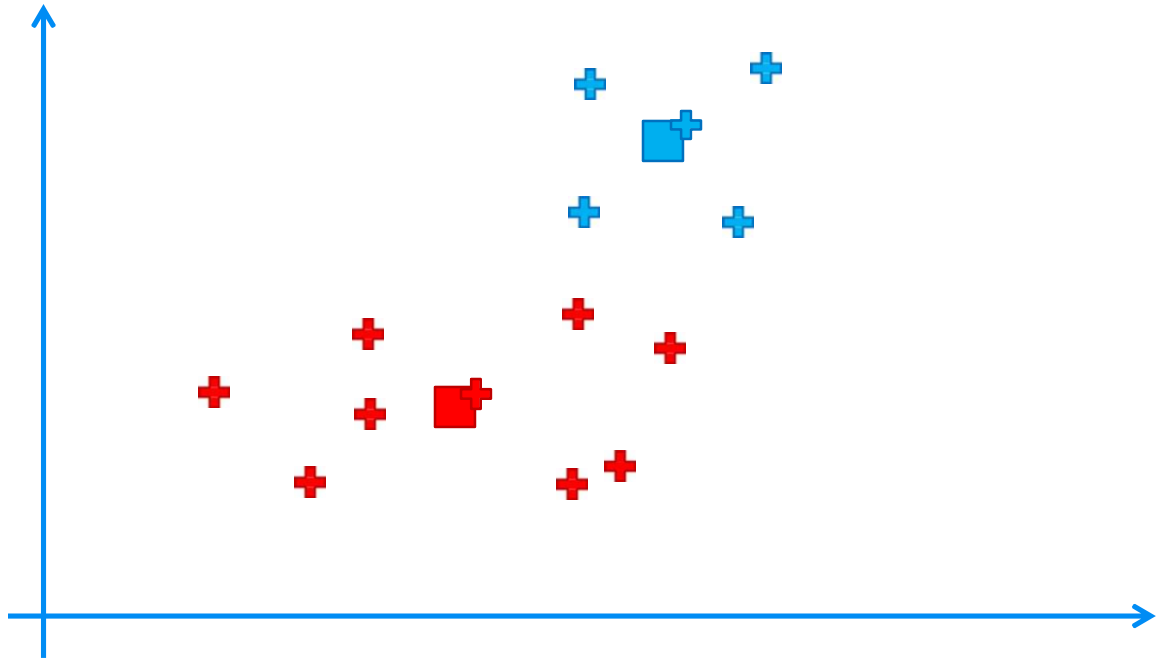
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

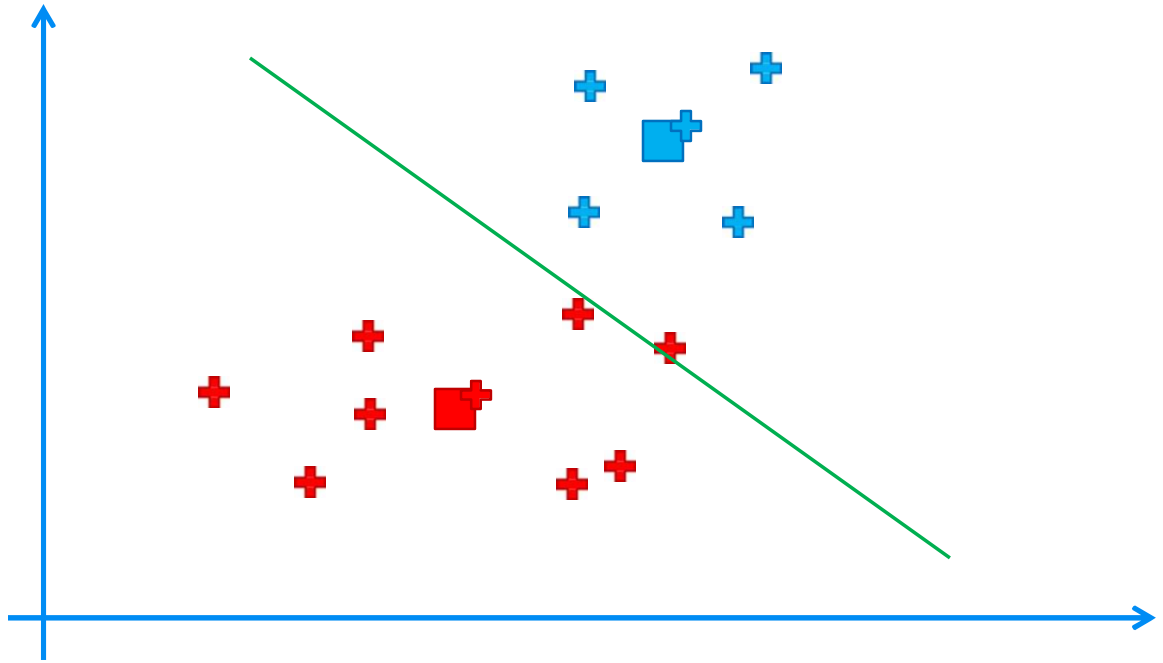
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

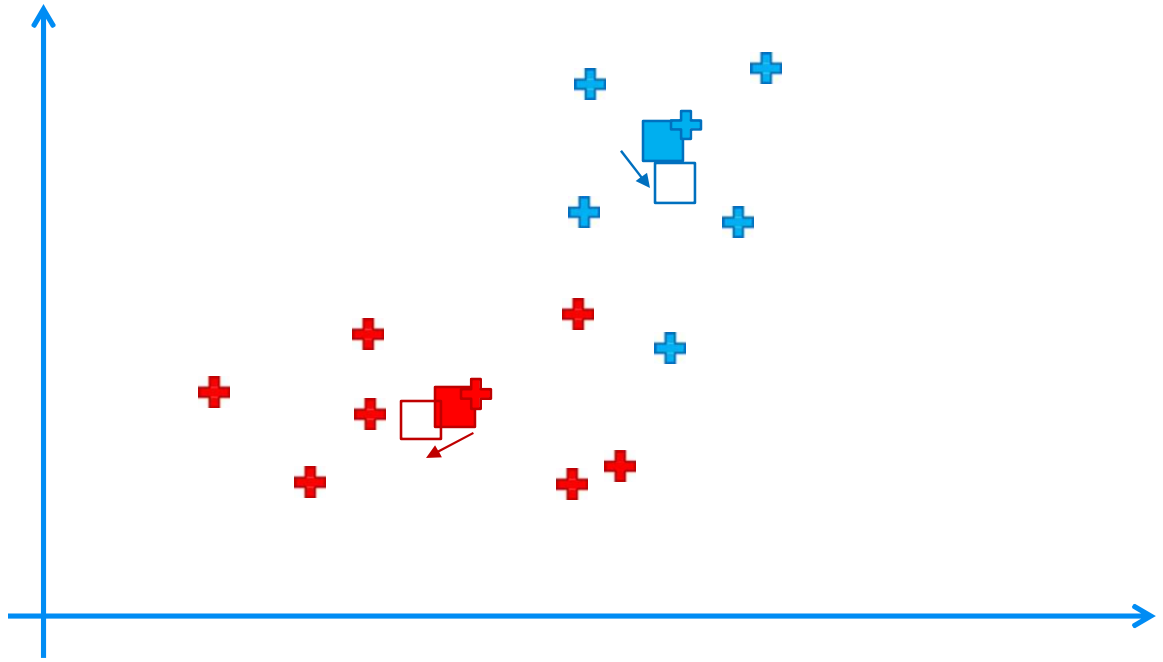
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

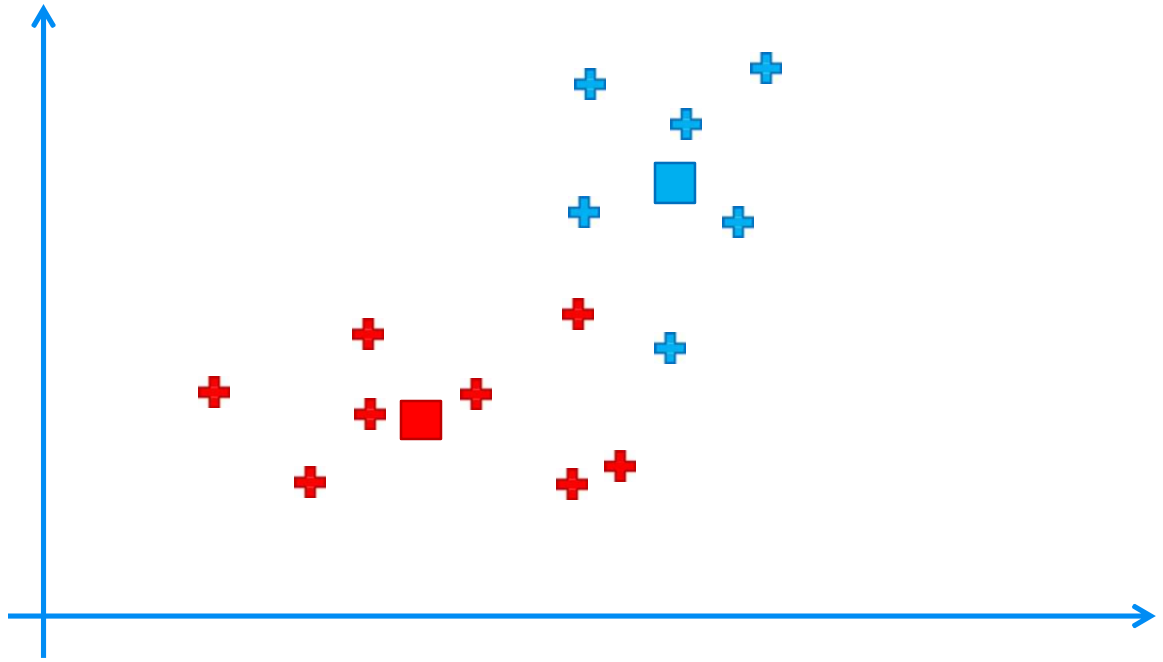
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

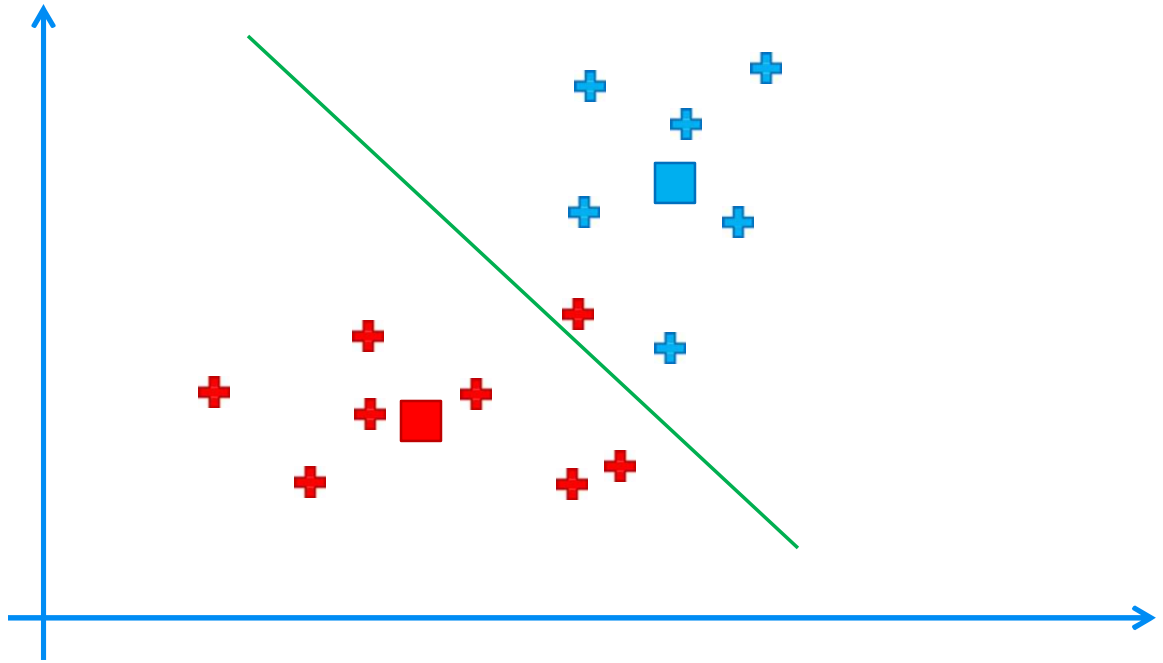
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

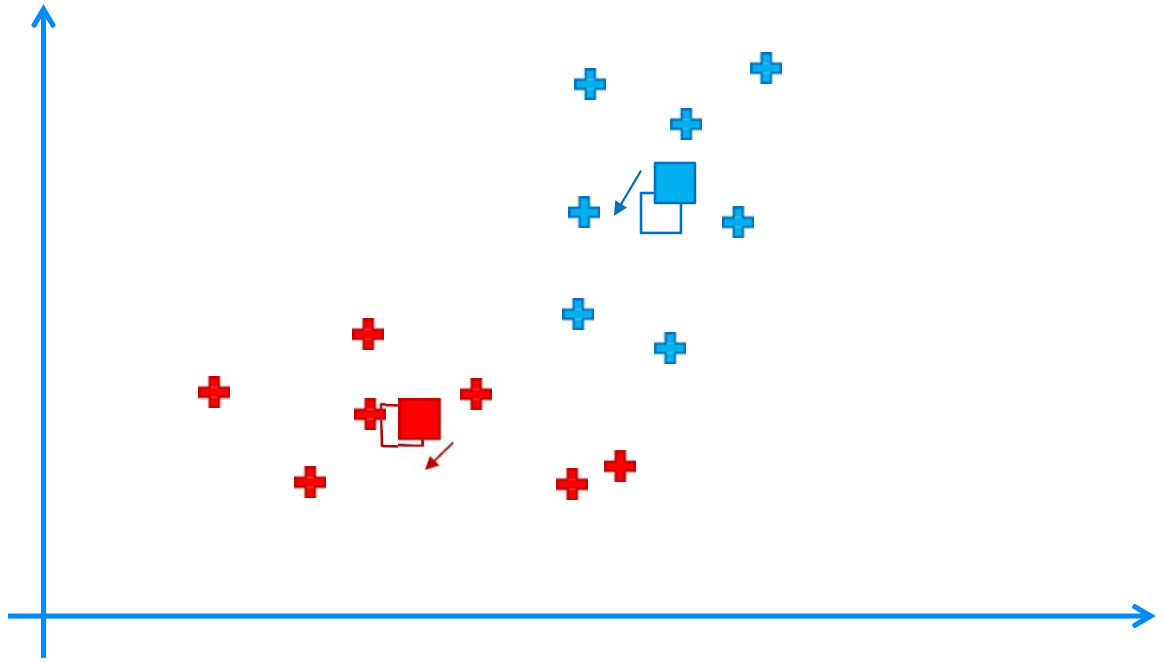
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

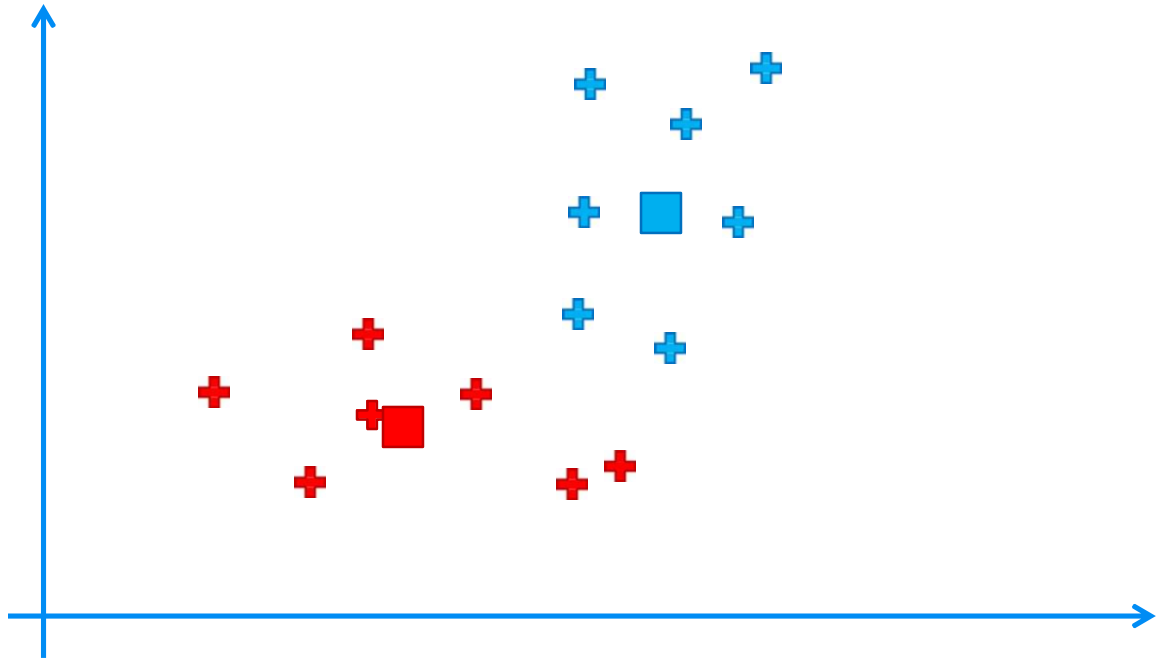
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

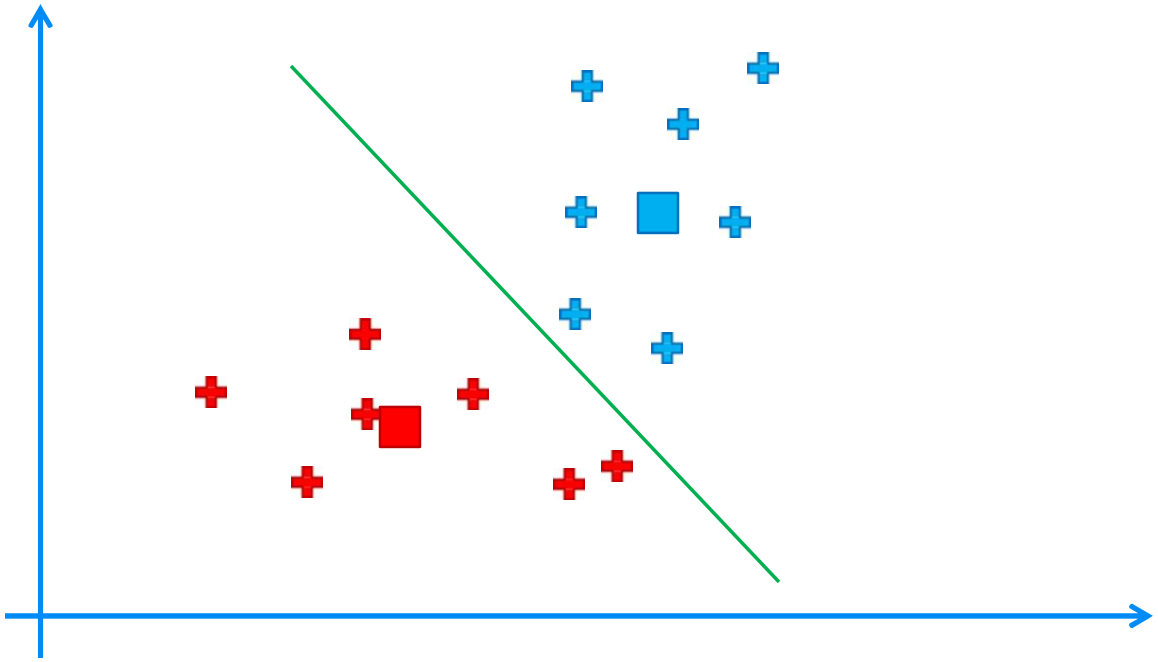
- Reassign
- Calculate the centroid
- Move the centroid.



Modelling – Clustering – K-Means

And now we repeat the same steps in a loop:

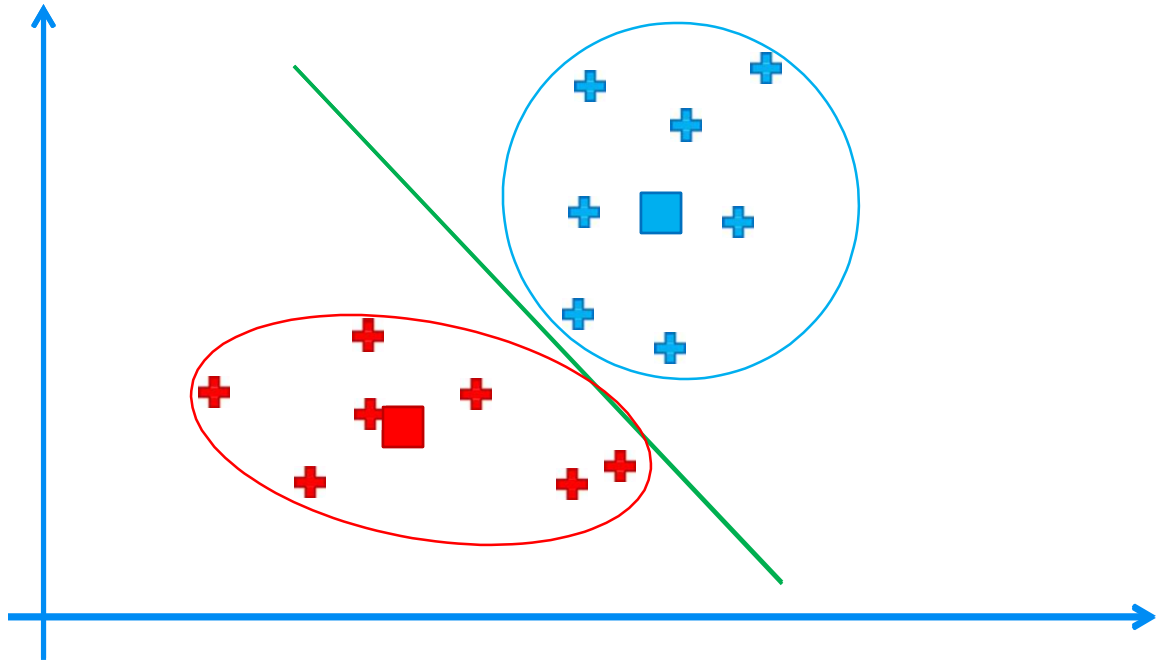
- Reassign
- Calculate the centroid
- Move the centroid.



Until the centroids no longer move

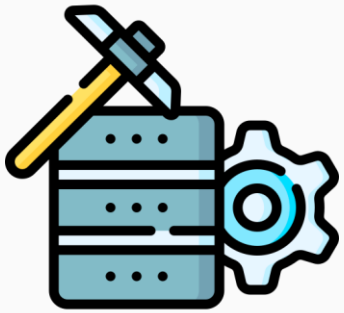
Modelling – Clustering – K-Means

And finally we have the final clusters!



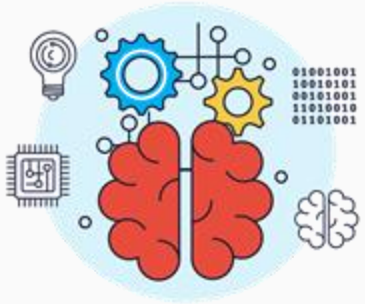
The Learning Process

Data Pre-processing



- ✓ Importing the data
- ✓ Data cleaning
- ✓ Splitting the dataset into training and test sets
- ✓ Features scaling

Modelling



- ✓ Building the model
- ✓ Training the model
- ✓ Making a prediction

Evaluation

- ✓ Calculating performance metrics
- ✓ Making a final assessment

Evaluation – Calculating performance metrics

A **confusion matrix** is a tool used to **evaluate the performance** of a classification model by comparing predicted values with actual values.

A table that summarizes the **number of correctly or incorrectly classified** instances for each class.

Truth		Predictions: Negative	Predictions: Positive
Truth	Negative	Prediction: **TRUE NEGATIVE** Truth:	Prediction: **FALSE POSITIVE** Truth:
Truth	Positive	Prediction: **FALSE NEGATIVE** Truth:	Prediction: **TRUE POSITIVE** Truth:

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Evaluation – Calculating performance metrics

A **confusion matrix** is a tool used to **evaluate the performance** of a classification model by comparing predicted values with actual values.

A table that summarizes the **number of correctly or incorrectly classified** instances for each class.

Truth \ Predictions	Negative	Positive
Negative	Prediction: **TRUE NEGATIVE** Truth:	Prediction: **FALSE POSITIVE** Truth:
Positive	Prediction: **FALSE NEGATIVE** Truth:	Prediction: **TRUE POSITIVE** Truth:

Metrics

Accuracy measures the proportion of correctly predicted instances among the total number of instances.

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

Precision indicates the proportion of correctly predicted positive instances out of all instances predicted as positive. It measures the model's ability to avoid false positives.

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

Sensitivity, also known as Recall, measures the proportion of actual positive instances that were correctly identified by the model. It highlights the model's ability to capture true positives.

$$\text{Sensitivity (Recall)} = \frac{TP}{TP + FN}$$

The **F1-Score** is the harmonic mean of Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Introduction to Python

A quick tour of Python, Jupyter, Colab, and Libraries

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

What is Python?

Python is a high-level programming language, known for its incredible simplicity and readability.

- **Versatile**: Used for web development, artificial intelligence, data analysis, automation, and much more.
- **Interpretable**: Code is executed line by line, making debugging easier.
- **Easy to Read**: Its clean syntax (which uses indentation) makes it almost look like English.

Why is so popular?

Clear and Simple Syntax - Indentation defines blocks of code, not parentheses. This forces you to write clean code.

Easy to Learn - It's often the first language taught to beginners because of its gentle learning curve.

Wide Community - A huge global community means there's always a tutorial, guide, or answer to a problem.

Immense Library Ecosystem - Its true strength: For any task, there's almost always a library that already does it.

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

Tools: Notebooks

For data analysis and AI, the most common environment is not a simple file, but a "notebook."

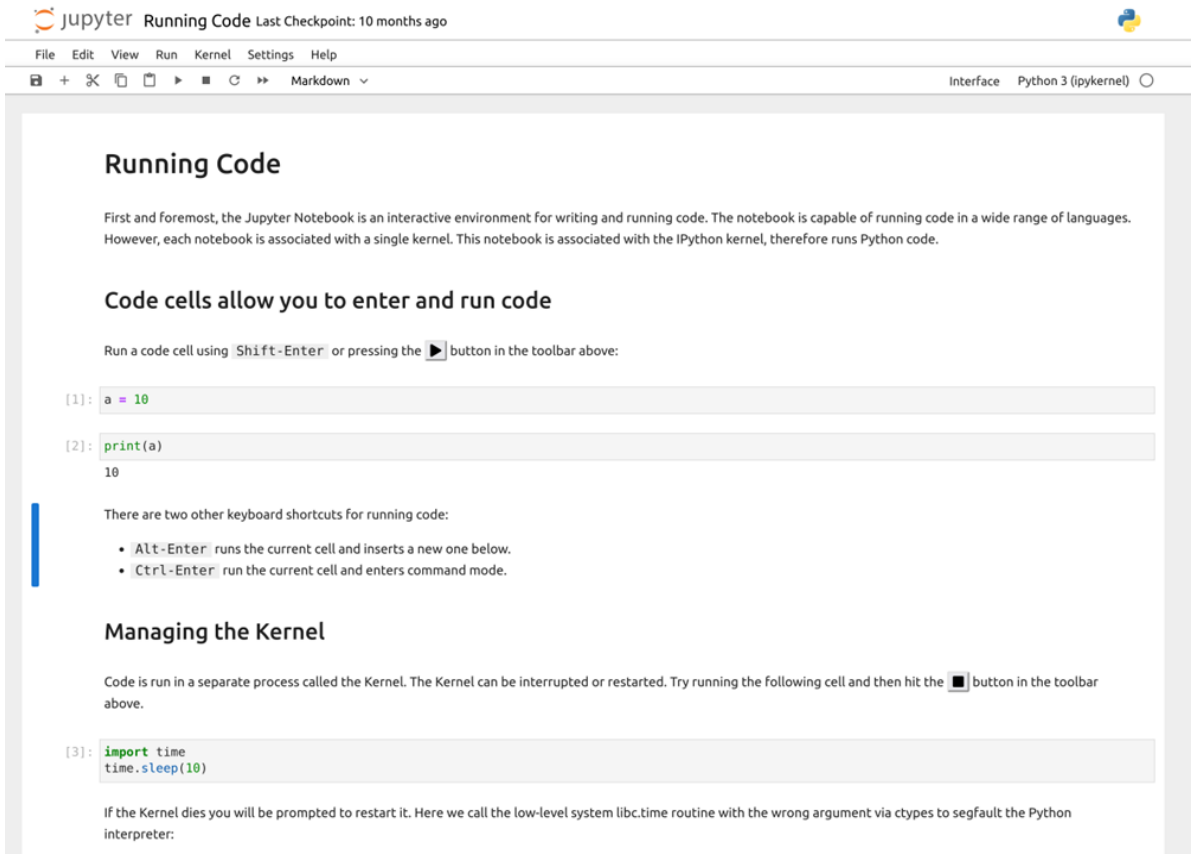
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

What is Jupyter Notebook?

Jupyter is an open-source web application that allows you to create and share interactive documents.

It is structured in cells:

- **Code Cells**: These contain Python code that can be executed block by block, displaying the output immediately.
- **Text Cells (Markdown)**: These allow you to write notes, insert images, formulas, and graphs alongside the code.
- It is the **standard tool for data exploration** and **teaching**.

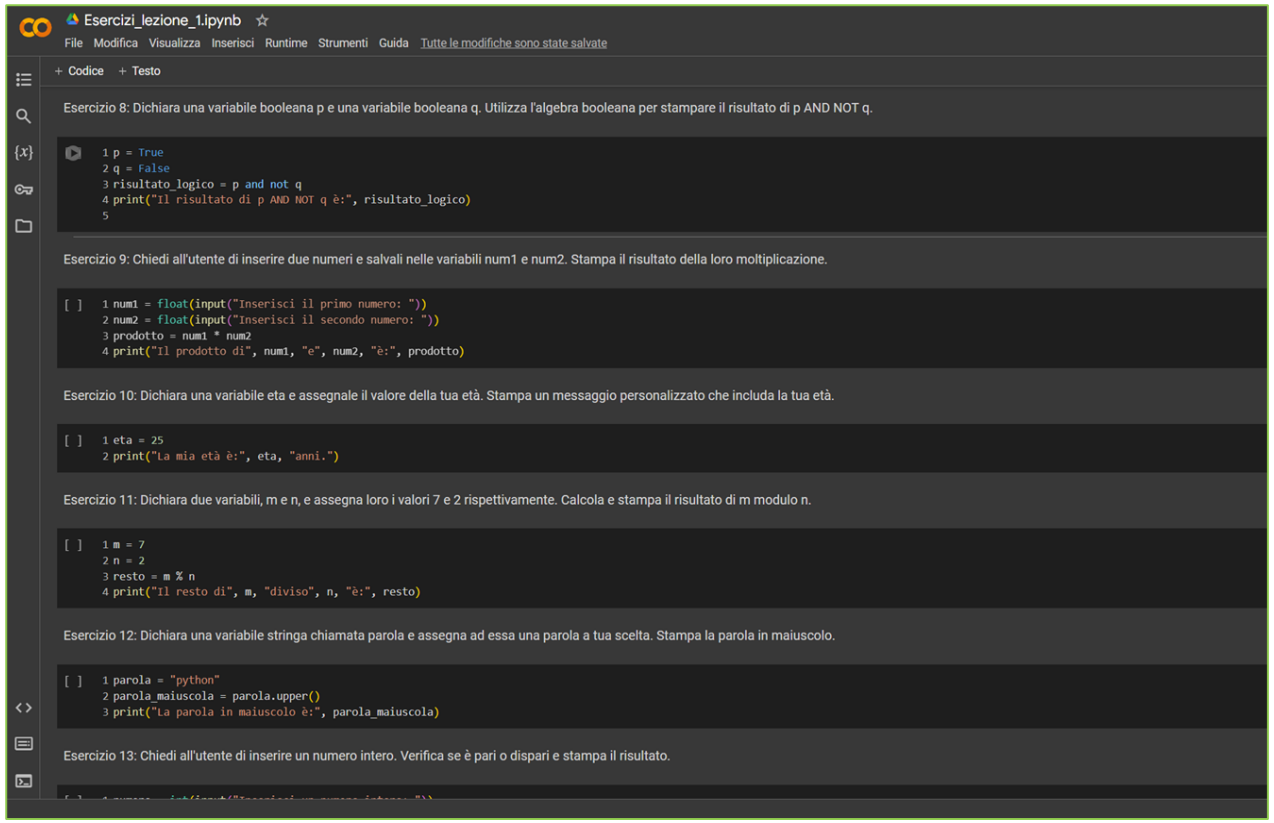


What is Google Colab?

In short: *Google Colab is a Jupyter Notebook in the cloud.*

It offers huge advantages:

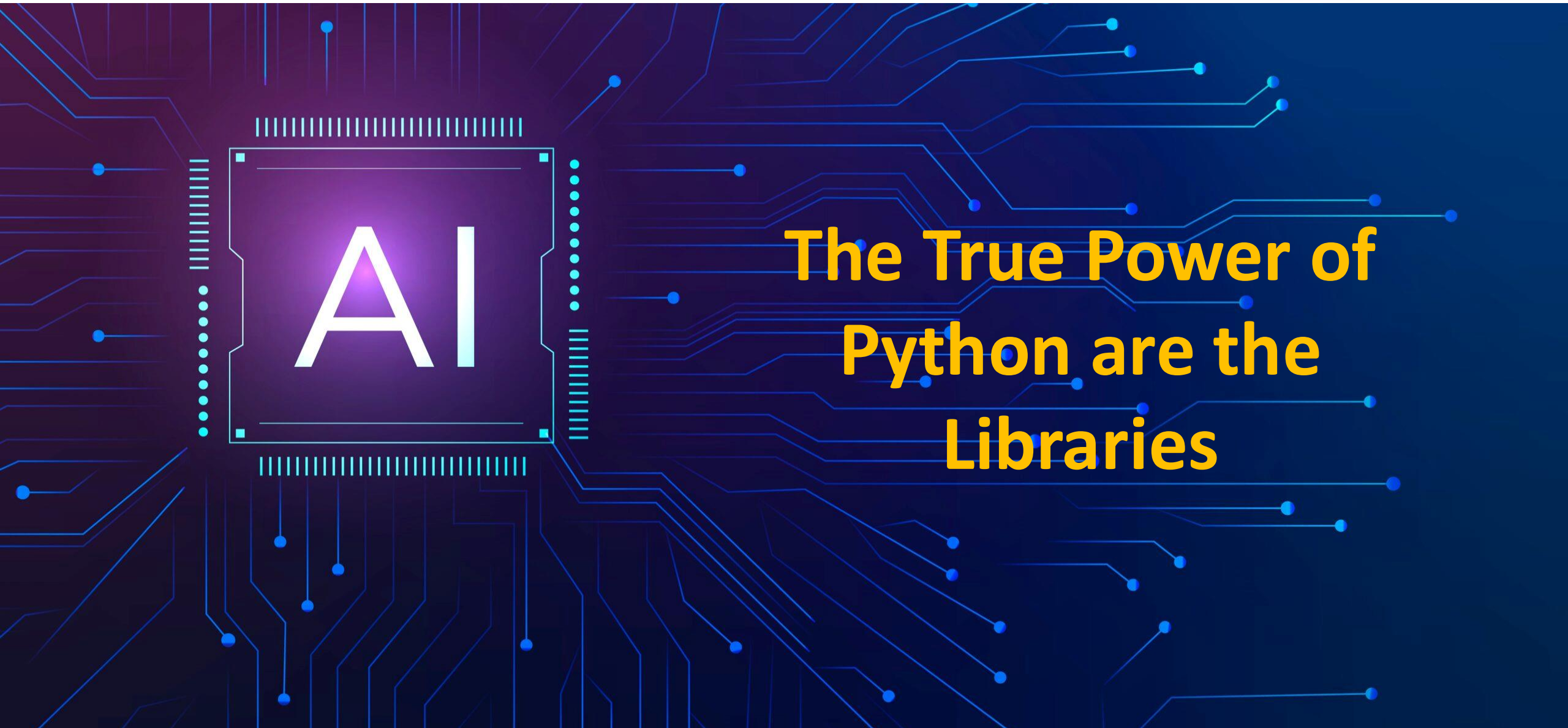
- **No Configuration**: It runs entirely in the browser. No installation required.
- **Free GPU/TPU Access**: Provides free computing power (GPU, TPU), essential for training machine learning models.
- **Collaborative**: It can be saved to Google Drive and shared and edited in real time, just like Google Docs.



Finanziato dall'Unione europea
NextGenerationEU

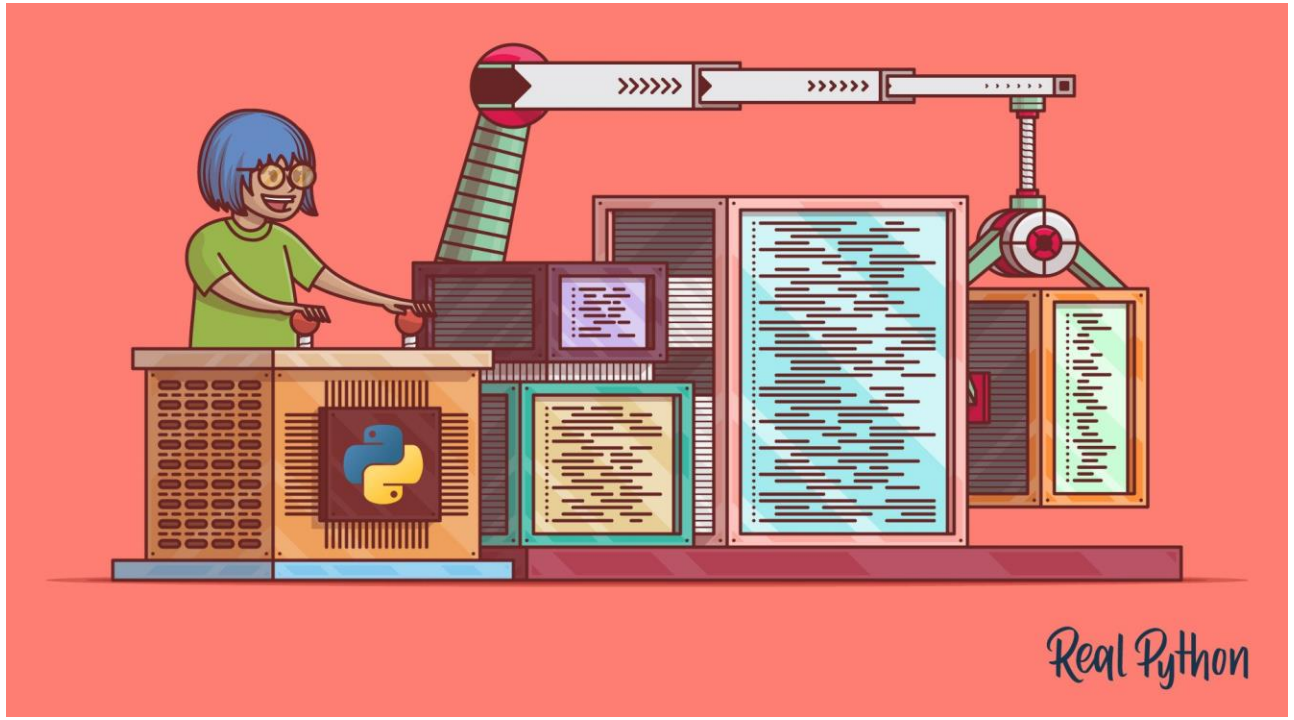
Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA



The True Power of Python are the Libraries

What are Libraries?



- **Ready-made "toolboxes"** - They are collections of code written by other programmers to solve common problems.
- **Modules** - A module is a single Python file (.py) filled with functions and classes you can import.
- **Libraries (or Packages)** - A library is a collection of multiple modules organized for a specific purpose (e.g., data analysis).
- **How are they used?** They are imported into your code with the import command (e.g., import pandas) to avoid having to reinvent the wheel.

The foundamental libraries for Data Science

NumPy

For numerical computations and arrays (matrices). It is the foundation of all scientific computation in Python.

Pandas

For analyzing and manipulating data in tables (called DataFrames). Perfect for reading CSV, Excel, and other files.

MatplotLib

Create graphs and visualizations. It allows you to create line graphs, bar graphs, histograms, and much more.

Advanced Libraries (Machine Learning)

Scikit-learn (sklearn)

It is the reference library for "classic" machine learning. It provides simple and efficient tools for:

- Classification
- Regression
- Clustering
- Data preprocessing

TensorFlow & PyTorch

These are the two main libraries for Deep Learning (Neural Networks), which is a specialized branch of Machine Learning used for complex tasks such as image and speech recognition.

In Summary

- **Python** - The programming language: easy, powerful, and versatile.
- **Jupyter & Google Colab** - Interactive development environments (on-premises or in the cloud) for analysis and experimentation.
- **Libraries (NumPy, Pandas, Sklearn, etc.)** - The true strength of Python. They provide specialized features that make Python the number one tool for Data Science and AI.

Hands on!