1.1 - Introduction to Machine Learning

Giovanni Della Lunga
giovanni.dellalunga@unibo.it

Introduction to Machine Learning for Finance

Bologna - February-March, 2025

Introduction to Machine Learning for Finance

- **Welcome to the Course!**
  - Instructor: Giovanni Della Lunga
  - Contact: giovanni.dellalunga@unibo.it
  - Course Duration: February-March 2025
- **Course Goals:**
  - Develop a solid understanding of Machine Learning (ML) fundamentals.
  - Explore ML applications in finance, including prediction and risk assessment.
  - Gain hands-on experience with Python tools for ML.

Introduction to Machine Learning for Finance

Lecture Overview

In this lesson, we will explore the foundational concepts of Machine Learning (ML).

Key Topics Covered

- What is Machine Learning? Definitions and core principles.
- Types of ML: Supervised, Unsupervised, Semi-supervised, and Reinforcement Learning.
- ML workflow: Data gathering, preprocessing, modeling, and evaluation.
- Essential ML concepts: Features, labels, cost functions, and optimization.
- Introduction to Python tools for ML: Pandas, scikit-learn, and Jupyter Notebooks.
- Using Pandas for financial data gathering.

Introduction to Machine Learning for Finance

Learning Objectives By the end of this lesson, students will:

- Understand the fundamental concepts of ML and its role in finance.
- Recognize different types of ML models (Supervised, Unsupervised, etc.).
- Learn how to gather and preprocess financial data using Python (Pandas, DataReader, yFinance).
- Gain an overview of the ML workflow from data collection to model optimization.

What is Machine Learning?

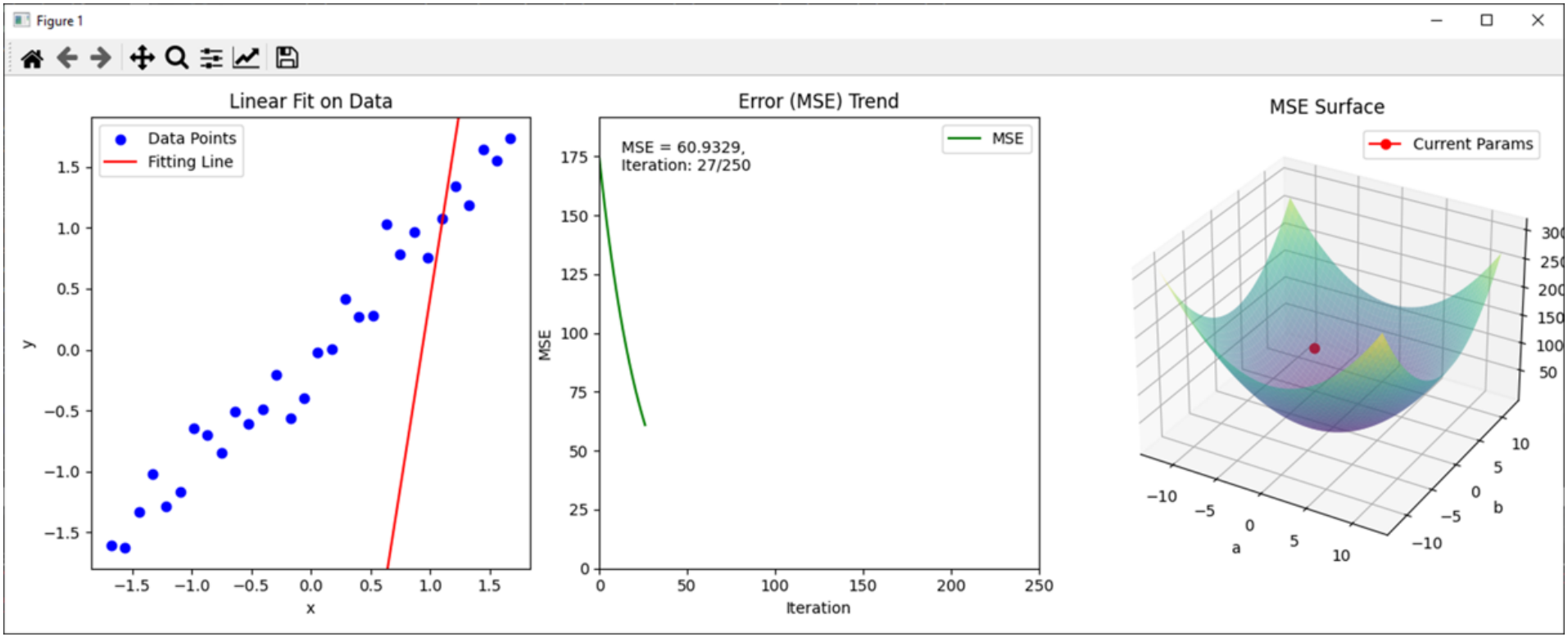
Basic Definitions

Defining Machine Learning

- Machine Learning is a scientific field within artificial intelligence (AI) that focuses on developing algorithms capable of learning from data.
- These algorithms identify patterns, relationships, or recurring motifs in data, which can include numerical information, textual content, images, or statistical records.
- Essentially, ML enables systems to perform tasks, make predictions, and improve their performance over time without explicit programming.
- This process involves feeding data into algorithms that, through analysis, uncover underlying structures to predict outcomes, classify information, or group similar entities.

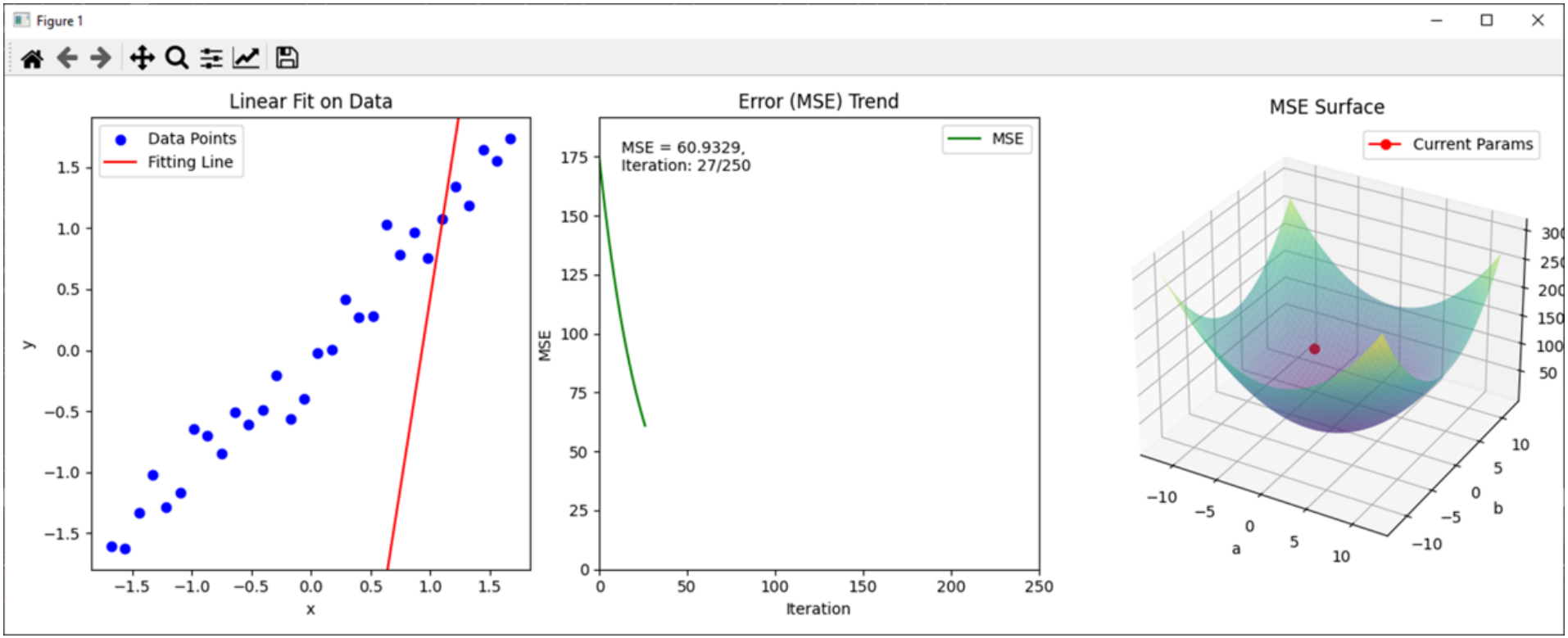
Defining Machine Learning

Machine Learning is a process where a **Model** learns from **Data** to make predictions or decisions. It compares its predictions to the actual outcomes, calculating the **Error**. Then, through **Optimization**, the model adjusts itself to reduce this error and improve its performance over time.



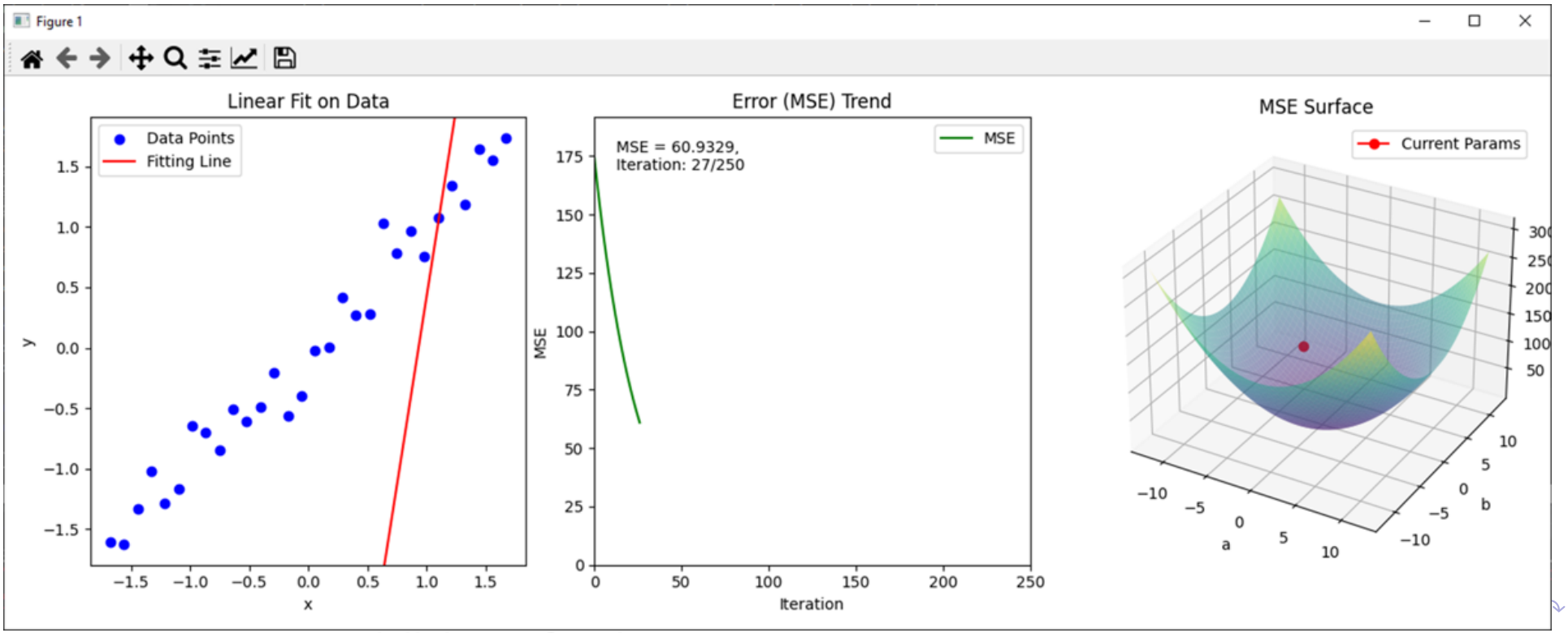
Defining Machine Learning

Data are the foundation of machine learning, as models rely on it to identify patterns and make predictions. In a dataset, **features** are the input variables that describe the data, while **labels** are the target outputs the model aims to predict or classify. Features provide the information, and labels define what the model is learning to predict.



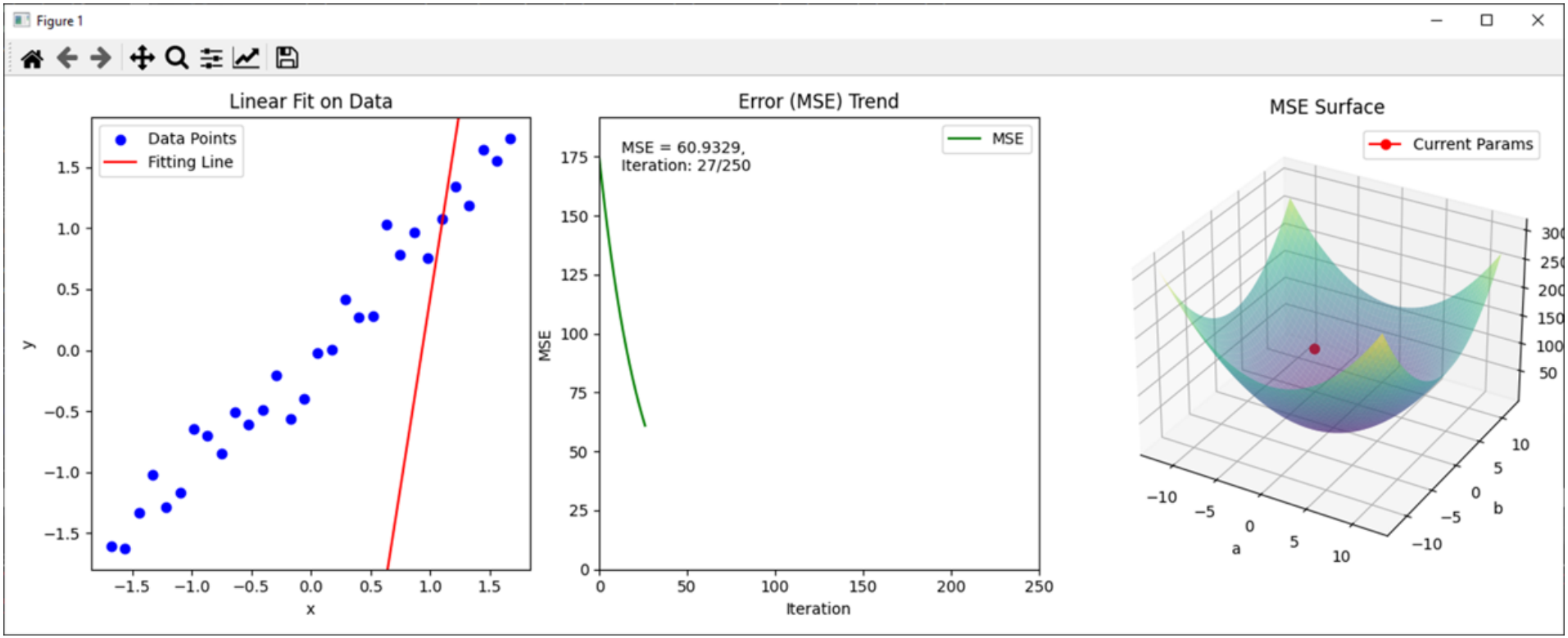
Defining Machine Learning

The **model** is the core of machine learning; it defines how data is processed to generate predictions. In simple linear regression, the model represents the relationship between input features and a target label as a straight line: $y = a + b * X$. Here, b (slope) and a (intercept) are parameters that the model learns from data. By adjusting these parameters during training, the model captures the underlying trend, enabling it to make accurate predictions on new data.



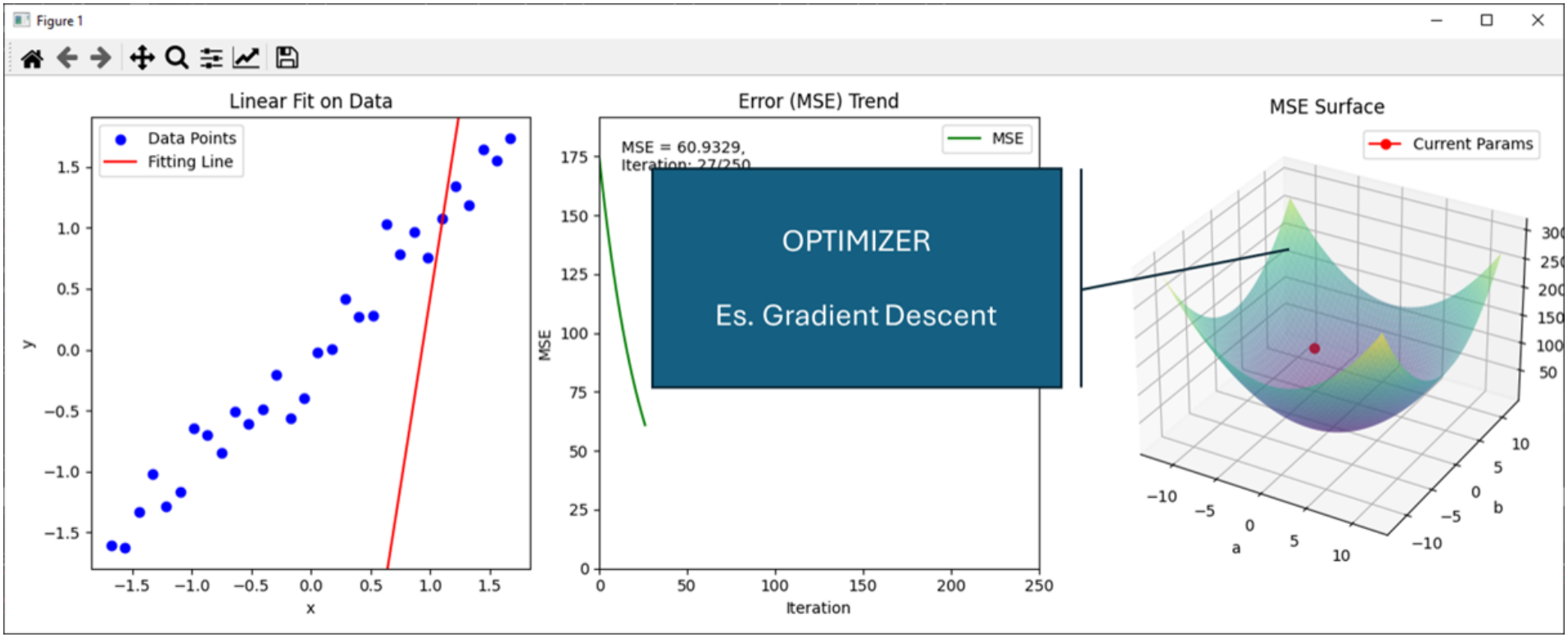
Defining Machine Learning

An **error measure** is essential in machine learning to quantify how far the model's predictions are from the actual values. It provides a way to evaluate the model's performance and guides improvements. By minimizing the error through optimization, we ensure the model learns effectively and generalizes well to new data.



Defining Machine Learning

An **optimizer** is used to adjust the model's parameters (e.g., weights in linear regression) to minimize the error. It works by iteratively updating the parameters in the direction that reduces the error, typically using methods like gradient descent. The optimizer calculates the gradient of the error with respect to the parameters and updates them step by step, ensuring the model improves its predictions over time.

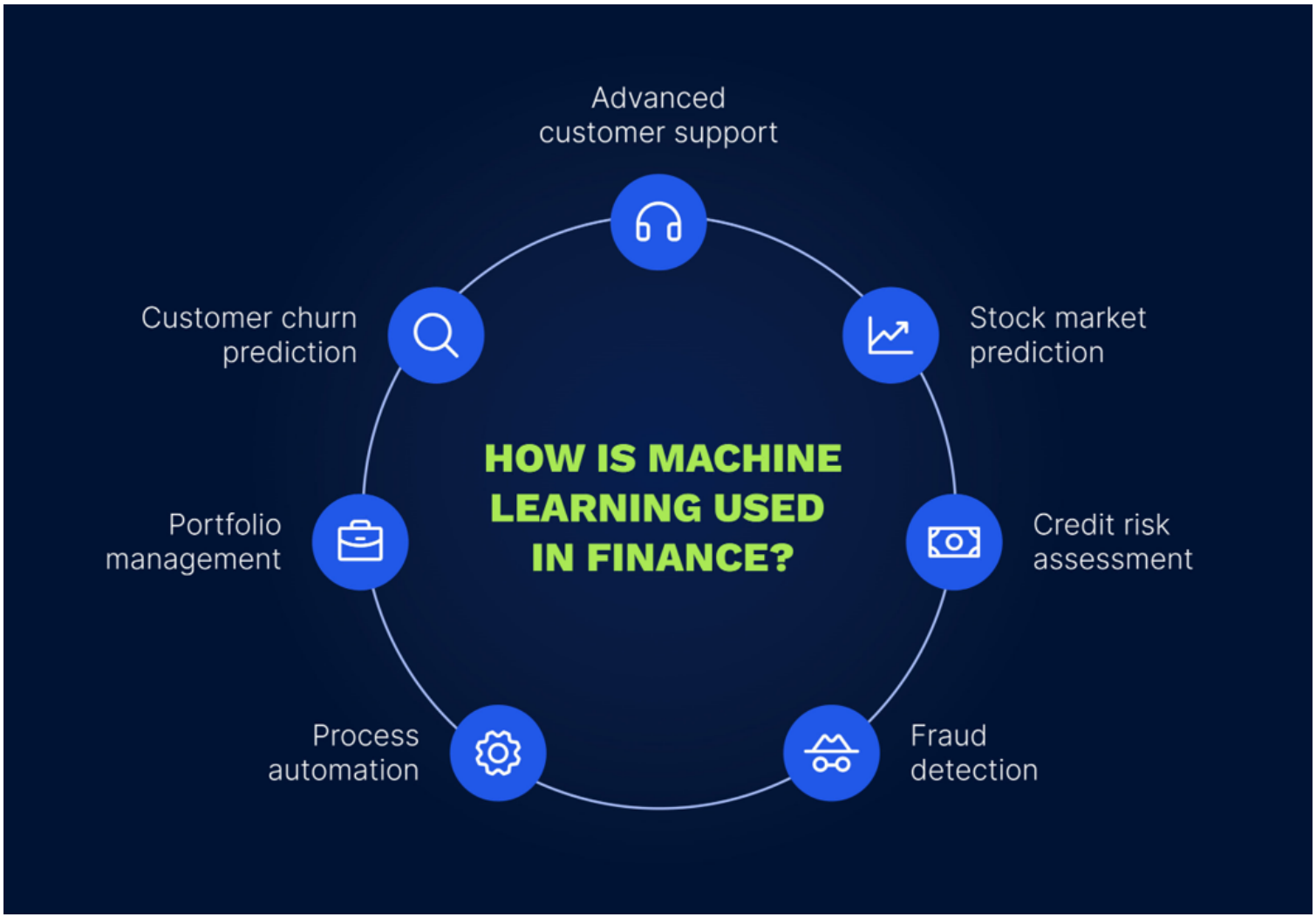


Defining Machine Learning

Click the link below to play the video: **Play Video**

Applications of Machine Learning

Machine Learning finds applications across diverse fields.

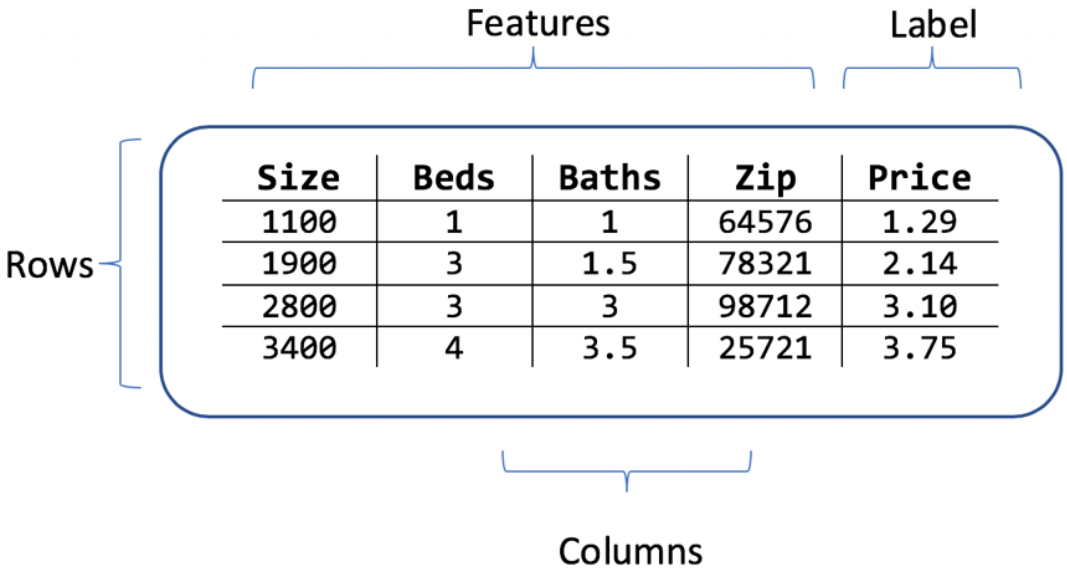


Subsection 1

Features and Labels

Features and Labels

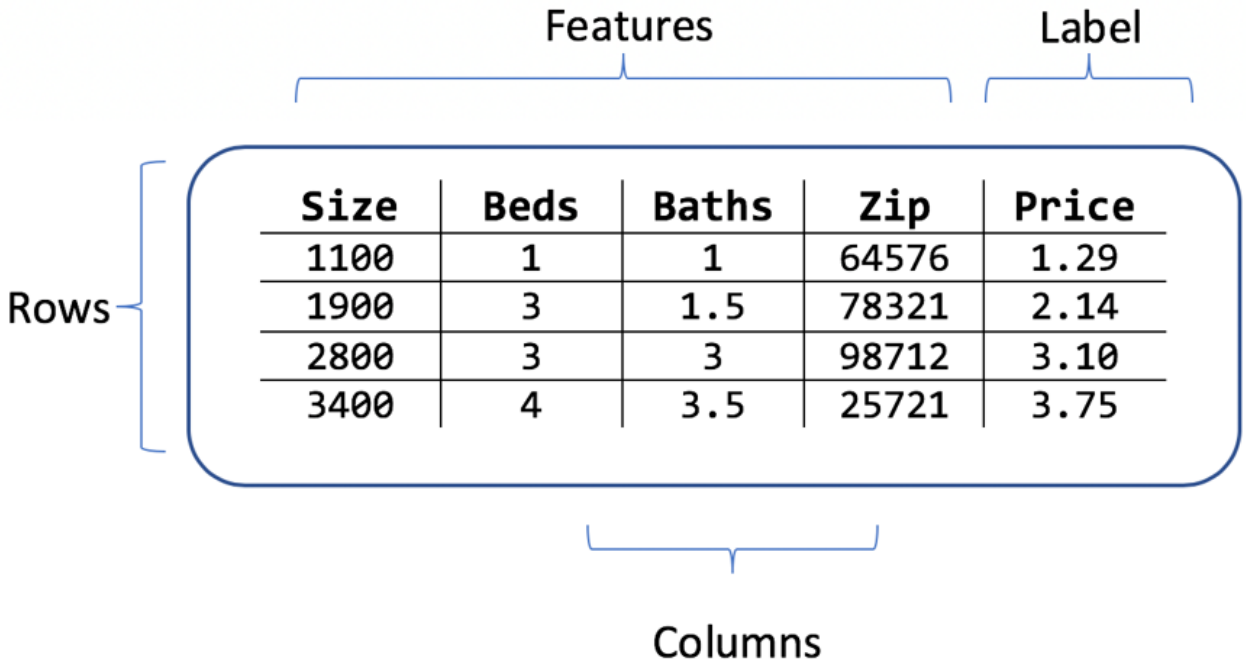
- The data for **ML Models** contains what are referred to as **features** and **labels**;
- *Labels* are the values of the target that is to be predicted;
- *Features* are the variables from which the predictions are to be made (if you are from statistics then think of it as an explanatory variable);



Size	Beds	Baths	Zip	Price
1100	1	1	64576	1.29
1900	3	1.5	78321	2.14
2800	3	3	98712	3.10
3400	4	3.5	25721	3.75

Features and Labels

- For example, when predicting the **price of a house**, the *features* could be the *square meters of living space*, *the number of bedrooms*, *the number of bathrooms*, *the size of the garage* and so on.
- The *label* would be *the house price*;



Subsection 2

Cost Functions

Cost Function

- In Machine Learning a **Cost Function** or loss function is used to represent how far away a mathematical model is from the real data ;
- One adjust the mathematical model usually by varying parameters within the model so as to minimize the const function;
- Let's take for example a very simple model of the form

$$y = \theta_0 + \theta_1 x$$

where the θs are the parameters that we want to find to give us the best fit to the data;

- Call this function $h_\theta(x)$ to emphasize the dependence on both the variable x and the two parameters θ_0 and θ_1;

Cost Function

- We want to measure how far away the data, the $y^{(n)}$s are from the function $h_\theta(x)$;
- A common way to do this is via the quadratic cost function

$$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} \left[h_\theta \left(x^{(n)} \right) - y^{(n)} \right]^2 \tag{1}$$

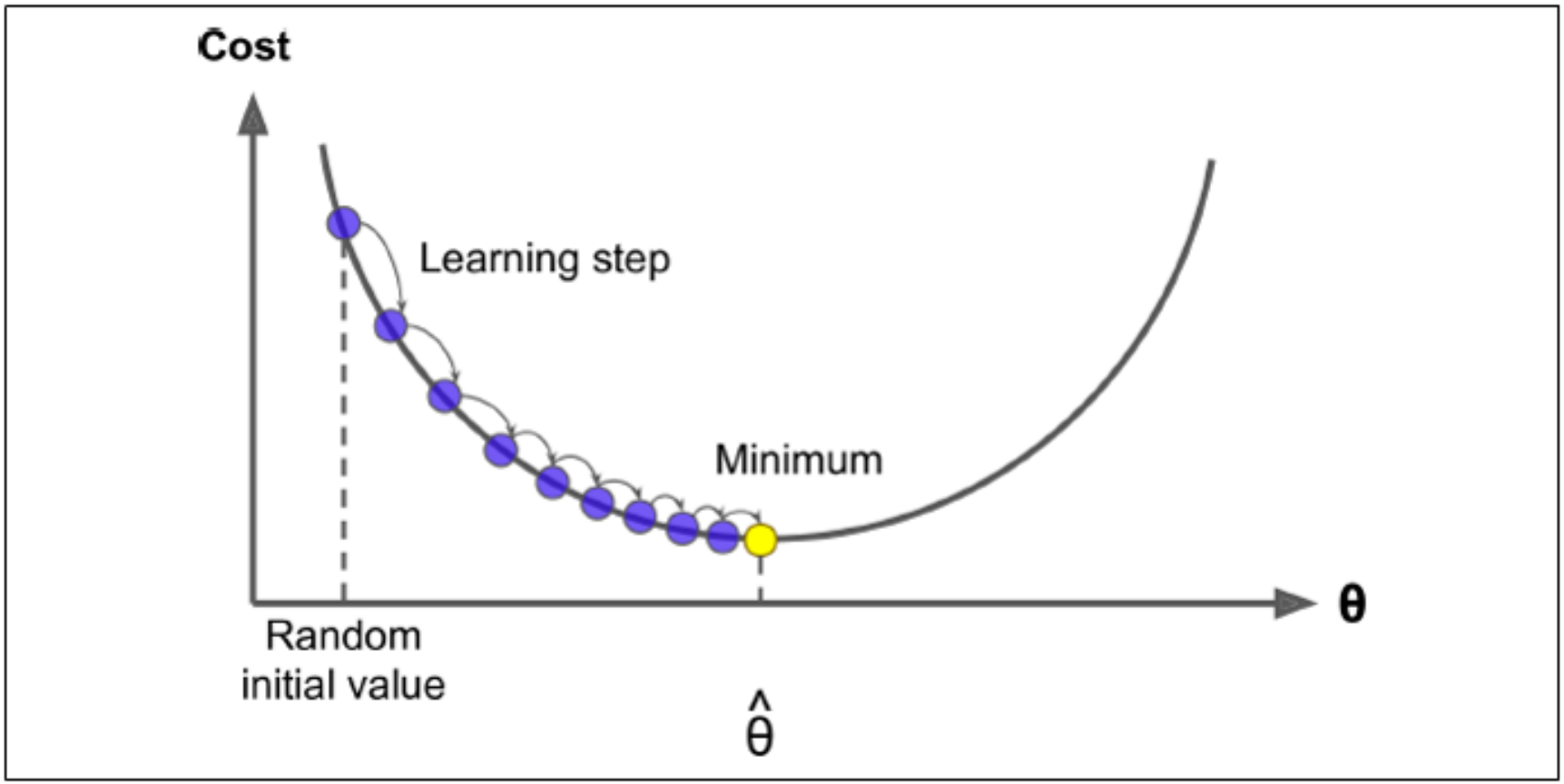
- We wants the parameters that minimize (**??**), almost always you are going to have to do this numerically;
- If we have a nice convex function then there is a numerical method that will converge to the solution, it is called **gradient descent**.

Gradient Descent

- Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems.
- Gradient Descent Algorithm measures the local gradient of the error function with regards to the parameter vector θ, and it goes in the direction of descending gradient.
- Once the gradient is zero, you have reached a minimum!
- Concretely, you start by filling θ with random values (this is called random initialization), and then you improve it gradually, taking one step at a time, each step attempting to decrease the cost function (e.g., the MSE), until the algorithm converges to a minimum

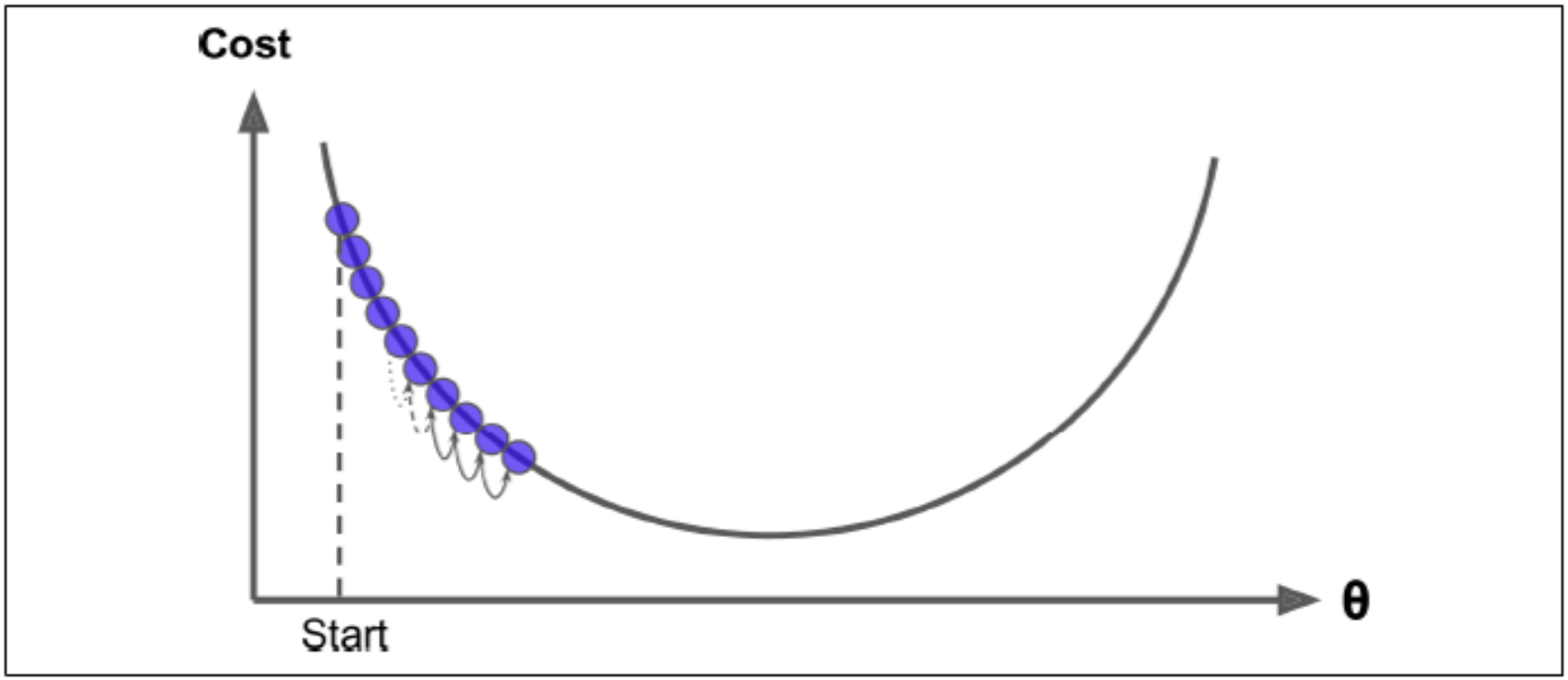
Gradient Descent

An important parameter in Gradient Descent is the size of the steps, determined by the learning rate hyperparameter.



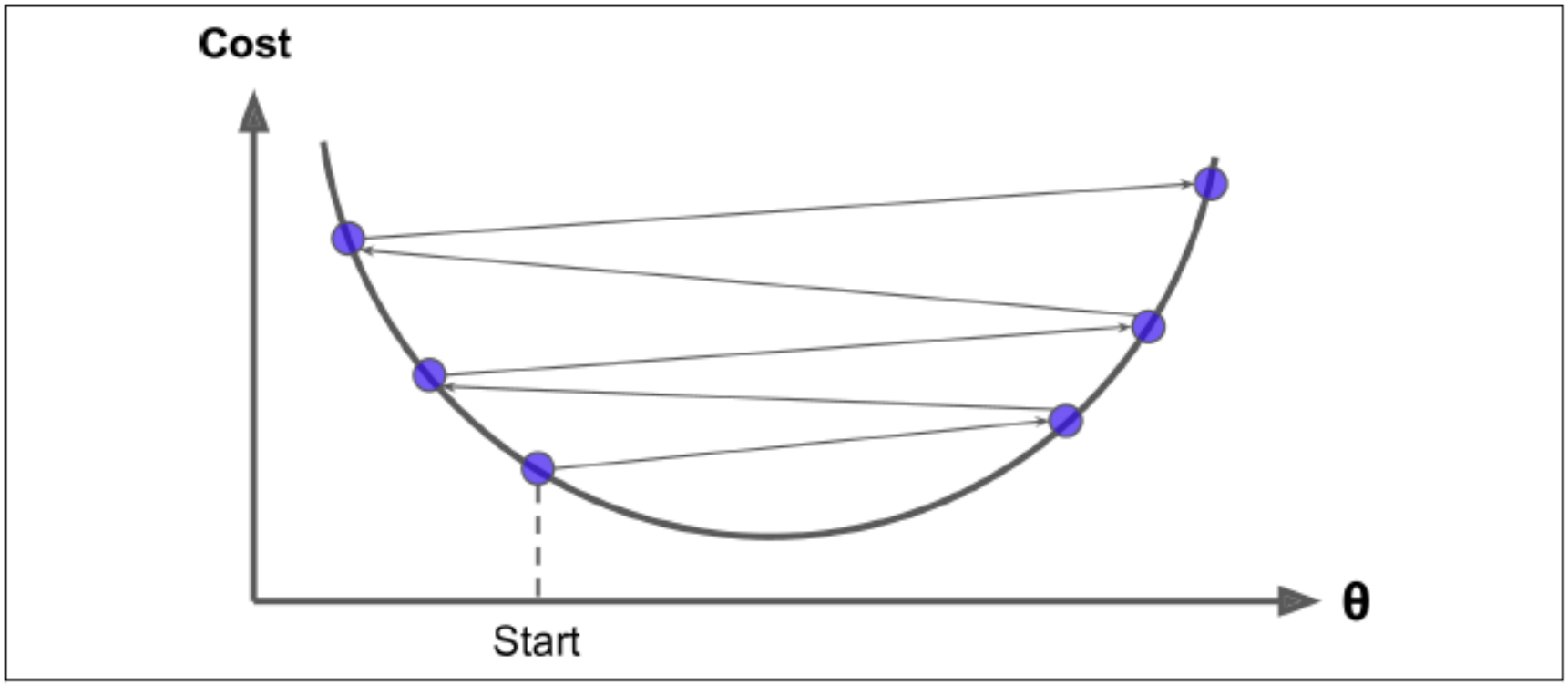
Gradient Descent

If the learning rate is too small, then the algorithm will have to go through many iterations to converge, which will take a long time...



Gradient Descent

... on the other hand, if the learning rate is too high, you might jump across the valley. This might make the algorithm diverge failing to find a good solution.



Gradient Descent

Having defined an appropriate learning rate, the scheme works as follow

- Start with an initial guess for each parameter θ_k;
- Move θ_k in the direction of the slope

$$\text{New } \theta_k = \text{Old } \theta_k - \beta \frac{\partial J}{\partial \theta_k}$$

where β is our learning rate;

In the above description of gradient descent we have used all of the data points simultaneously. This is called **batch gradient descent**

Gradient Descent

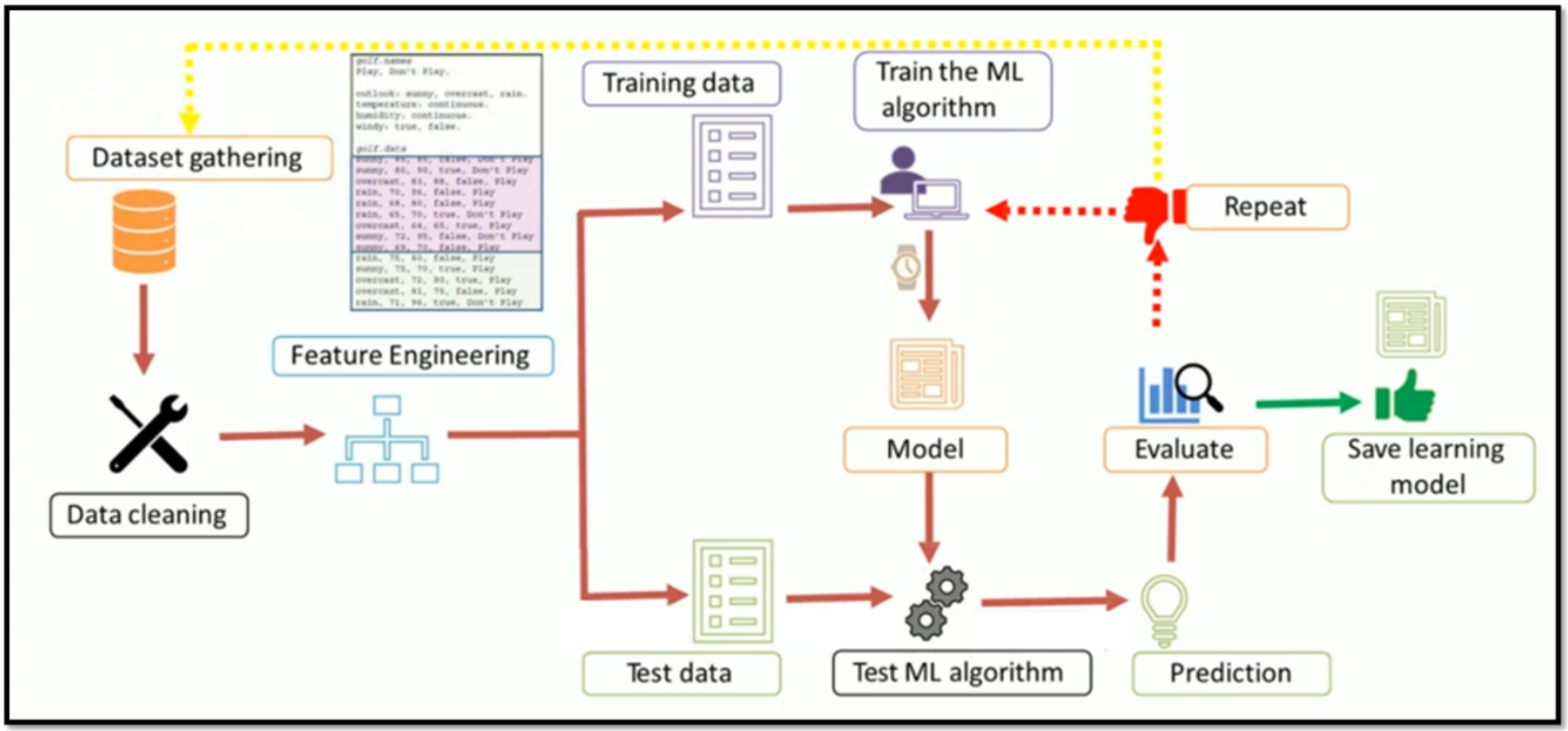
- Rather then use all of the data in the parameter updating we can use a technique called **stochastic gradient descent**;
- This technique is the same as the batch gradient descent except that you only update using **one** of the data points each time and this data point is chosen **randomly**;
- Especially in high-dimensional optimization problems this reduces the computational burden, achieving faster iterations in trade for a lower convergence rate;
- When the learning rates decrease with an appropriate rate, and subject to relatively mild assumptions, stochastic gradient descent converges almost surely to a global minimum when the objective function is convex or pseudoconvex, and otherwise converges almost surely to a local minimum;

The Machine Learning Process

General Workflow of a ML Application

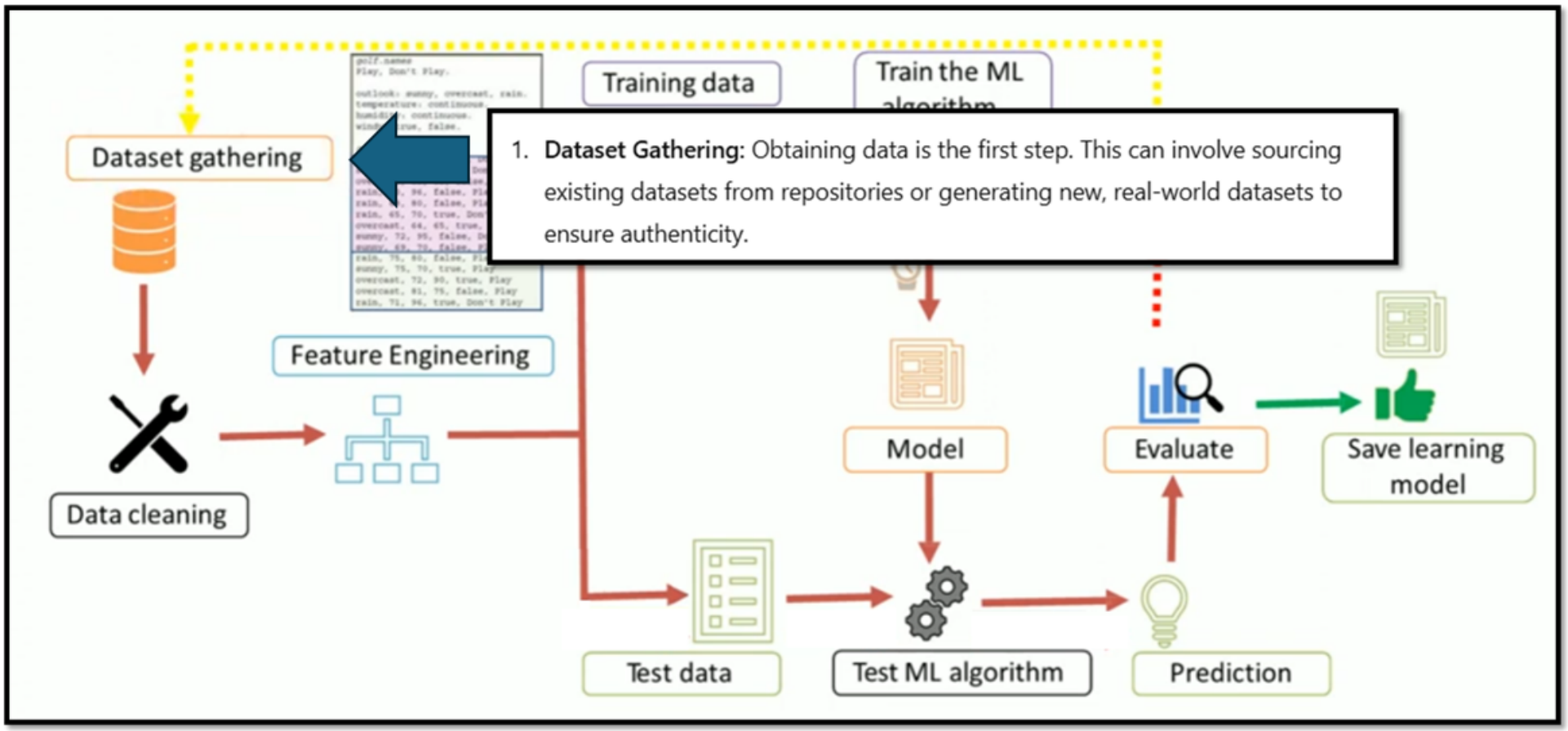
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



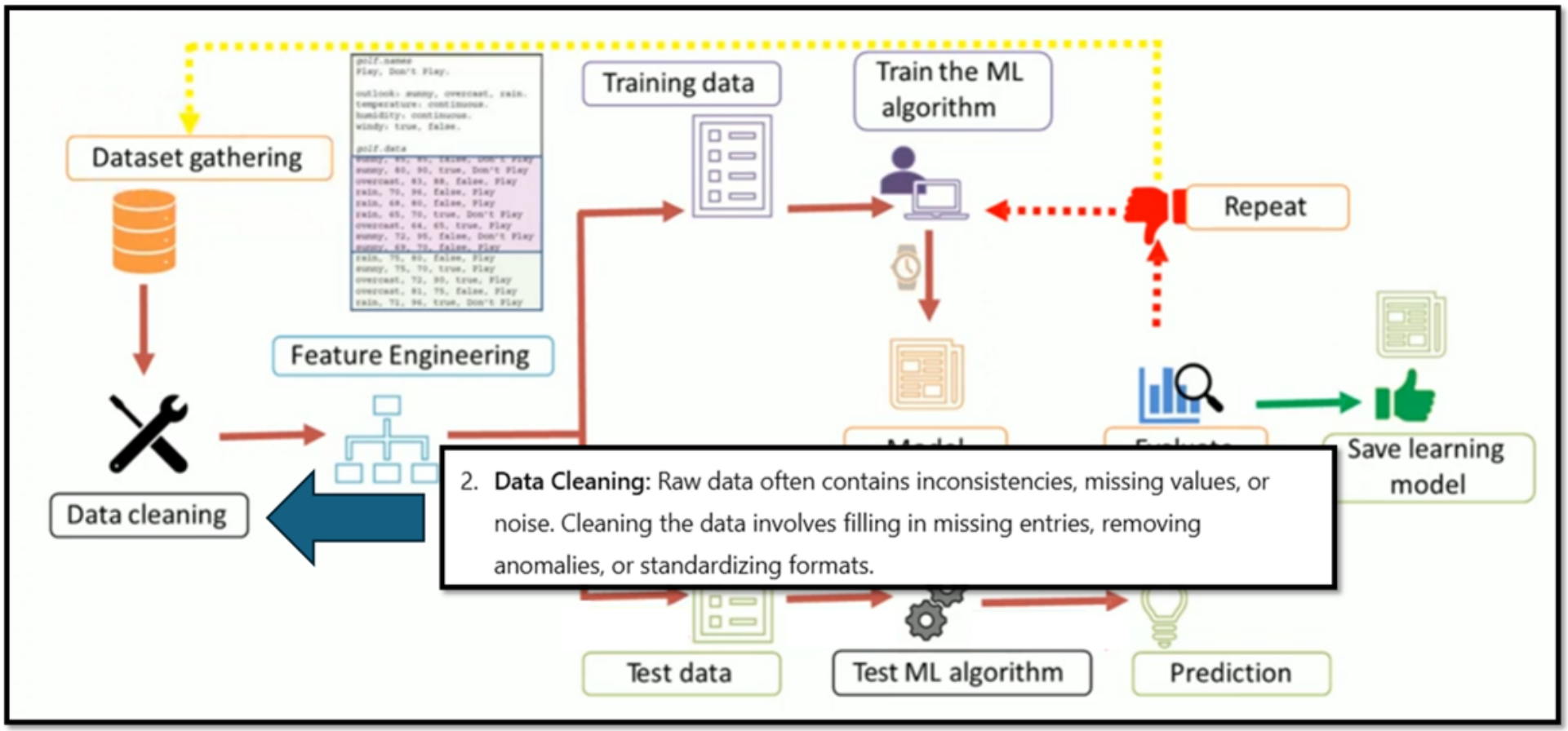
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



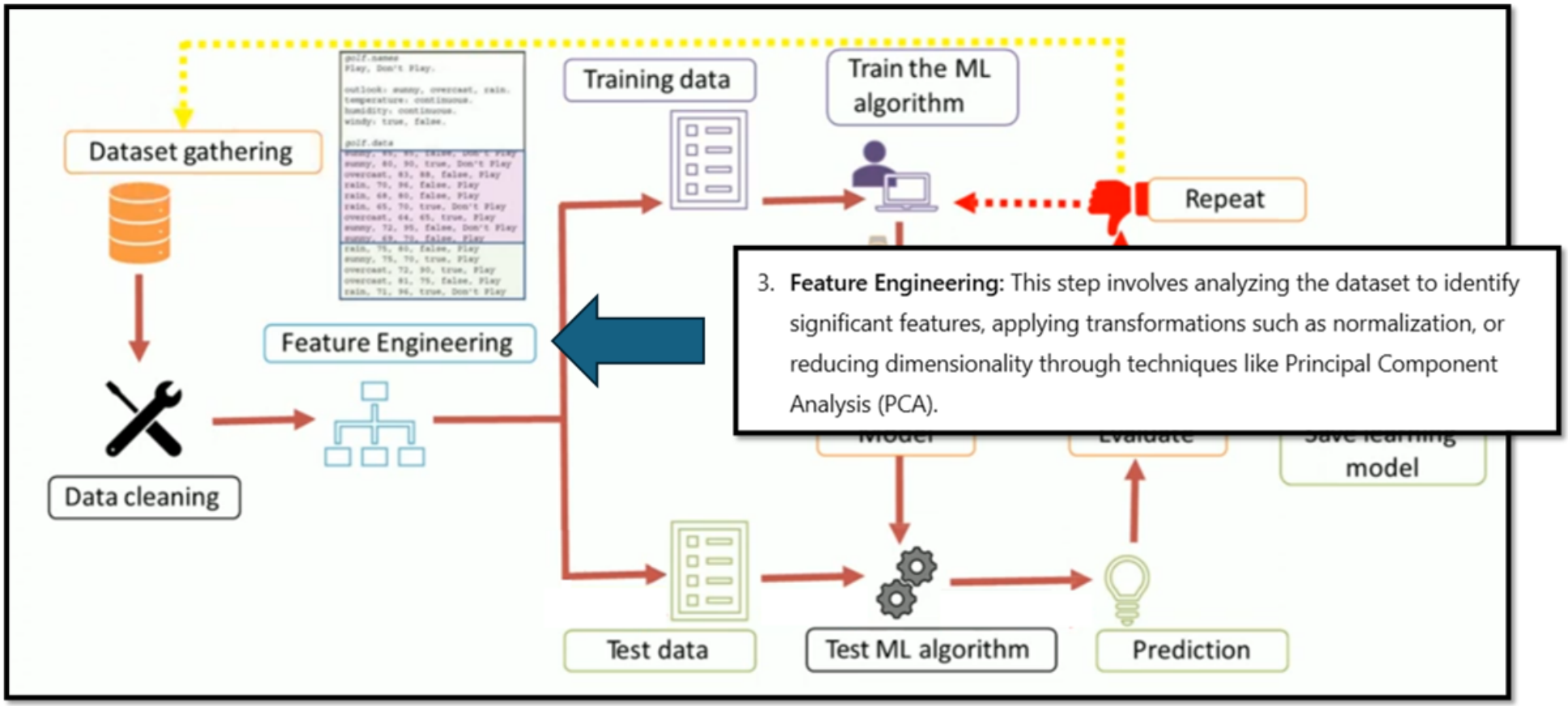
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



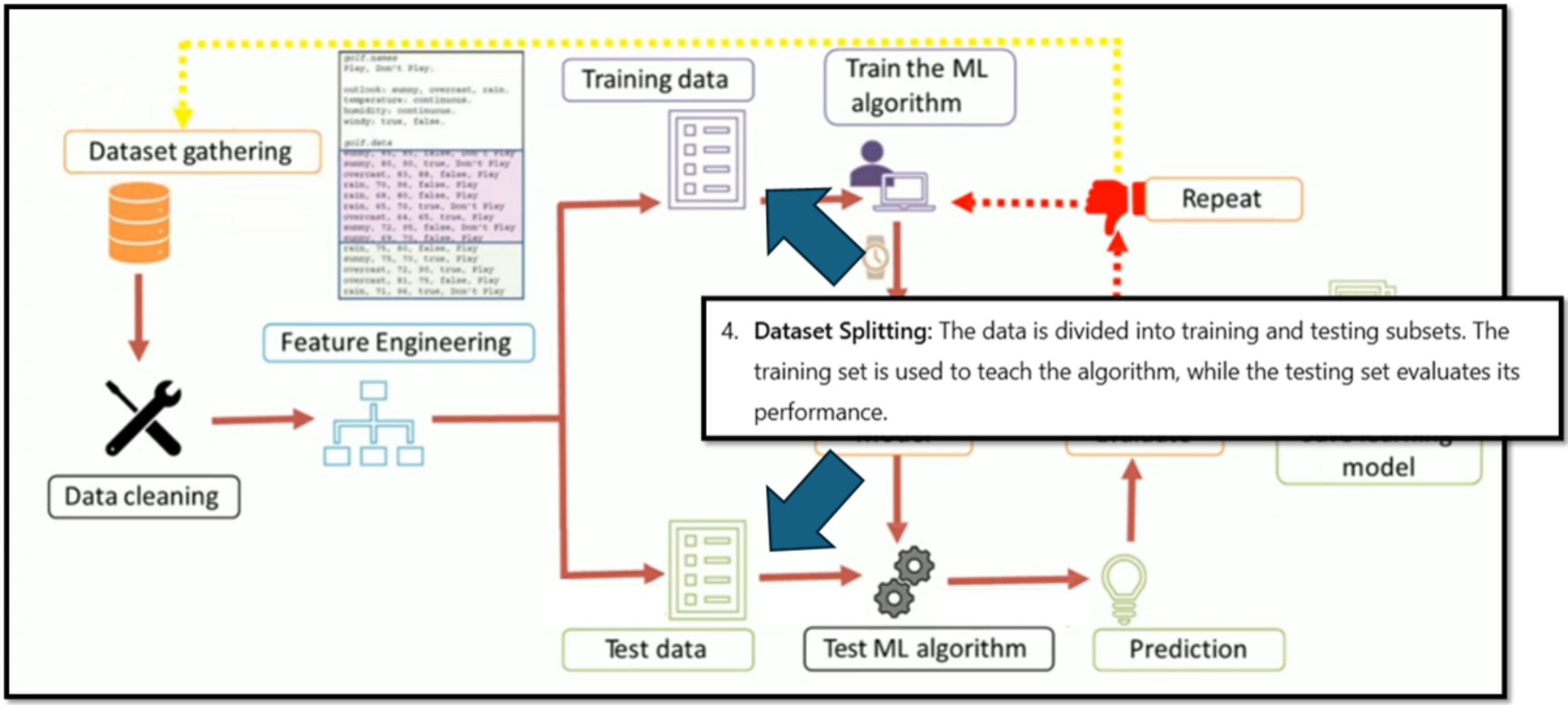
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



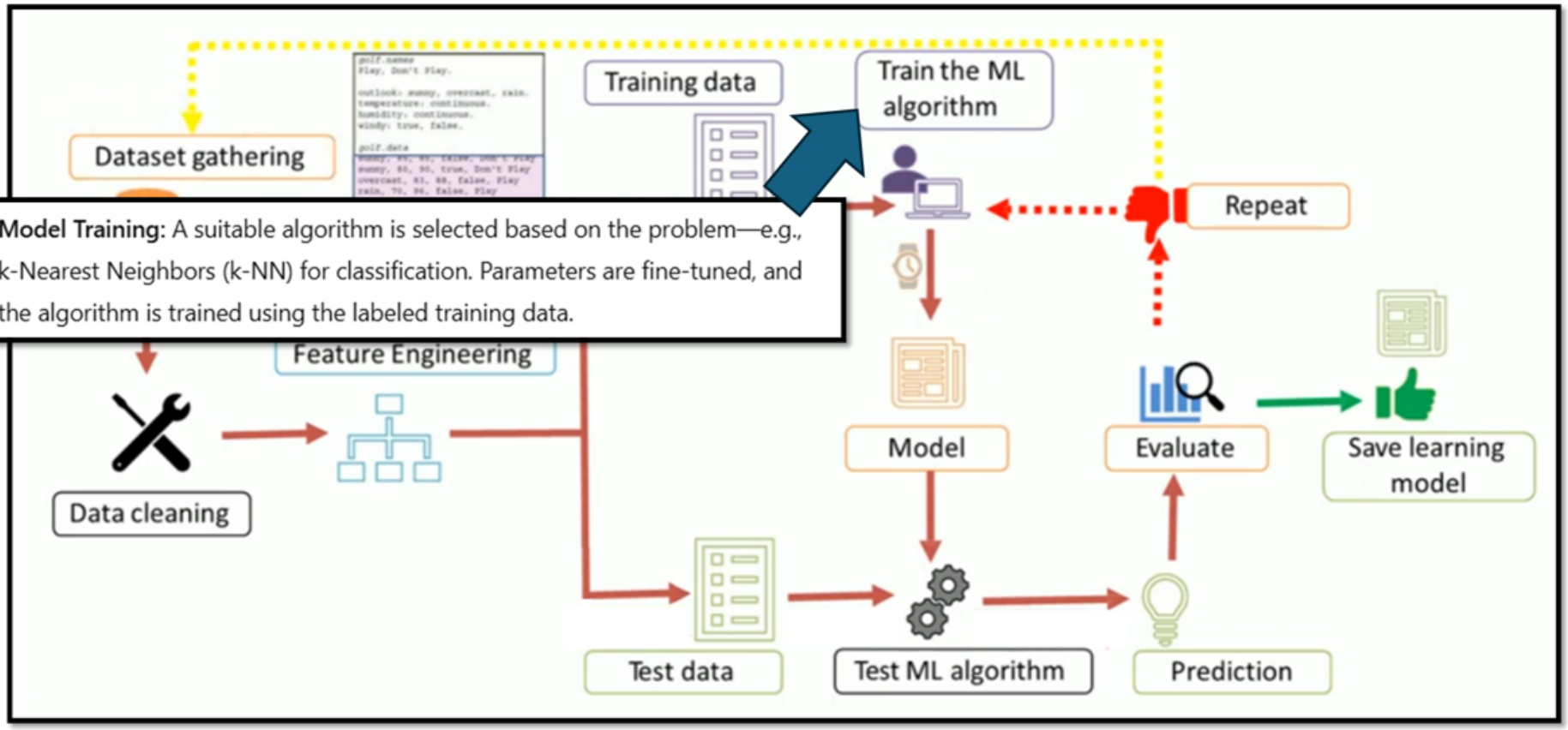
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



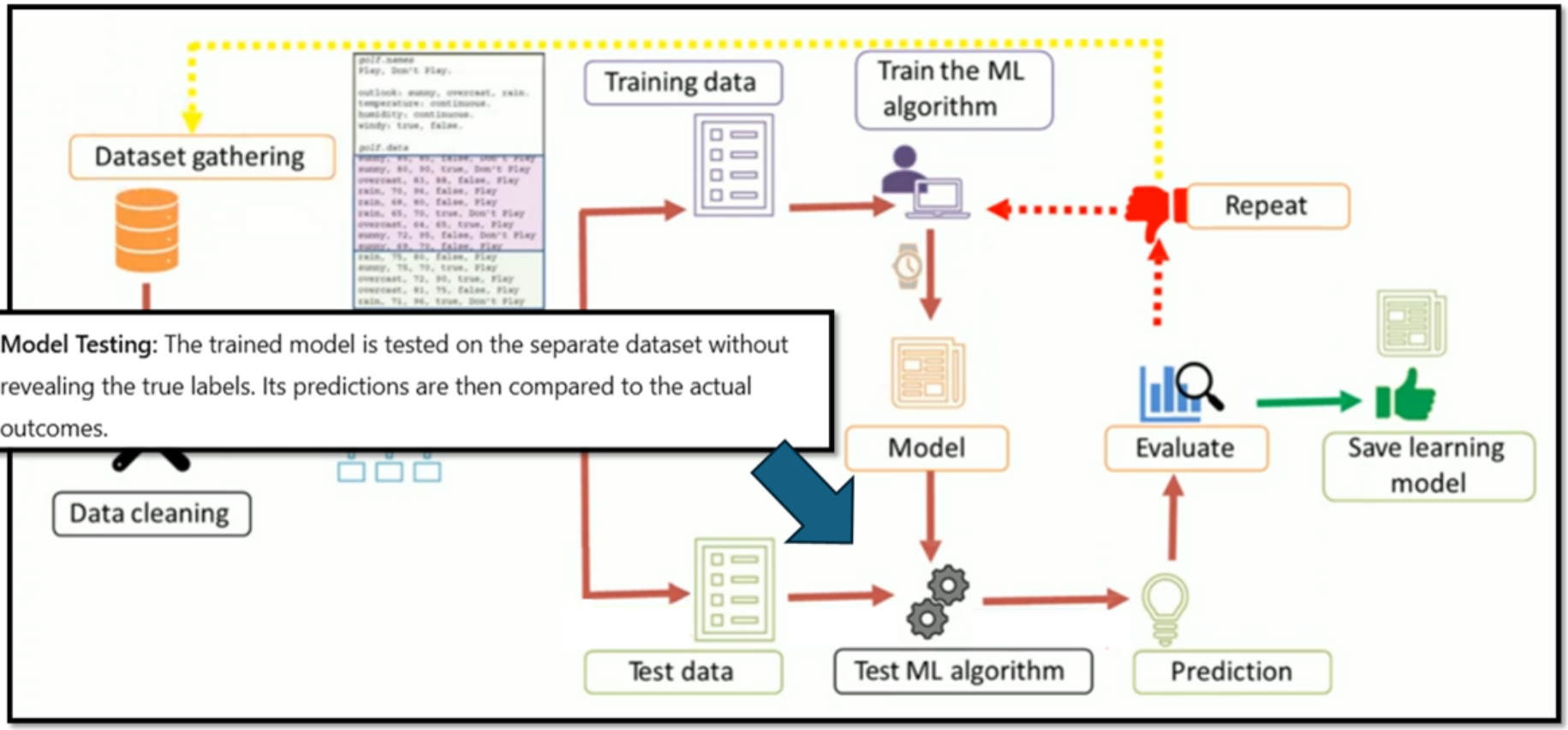
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



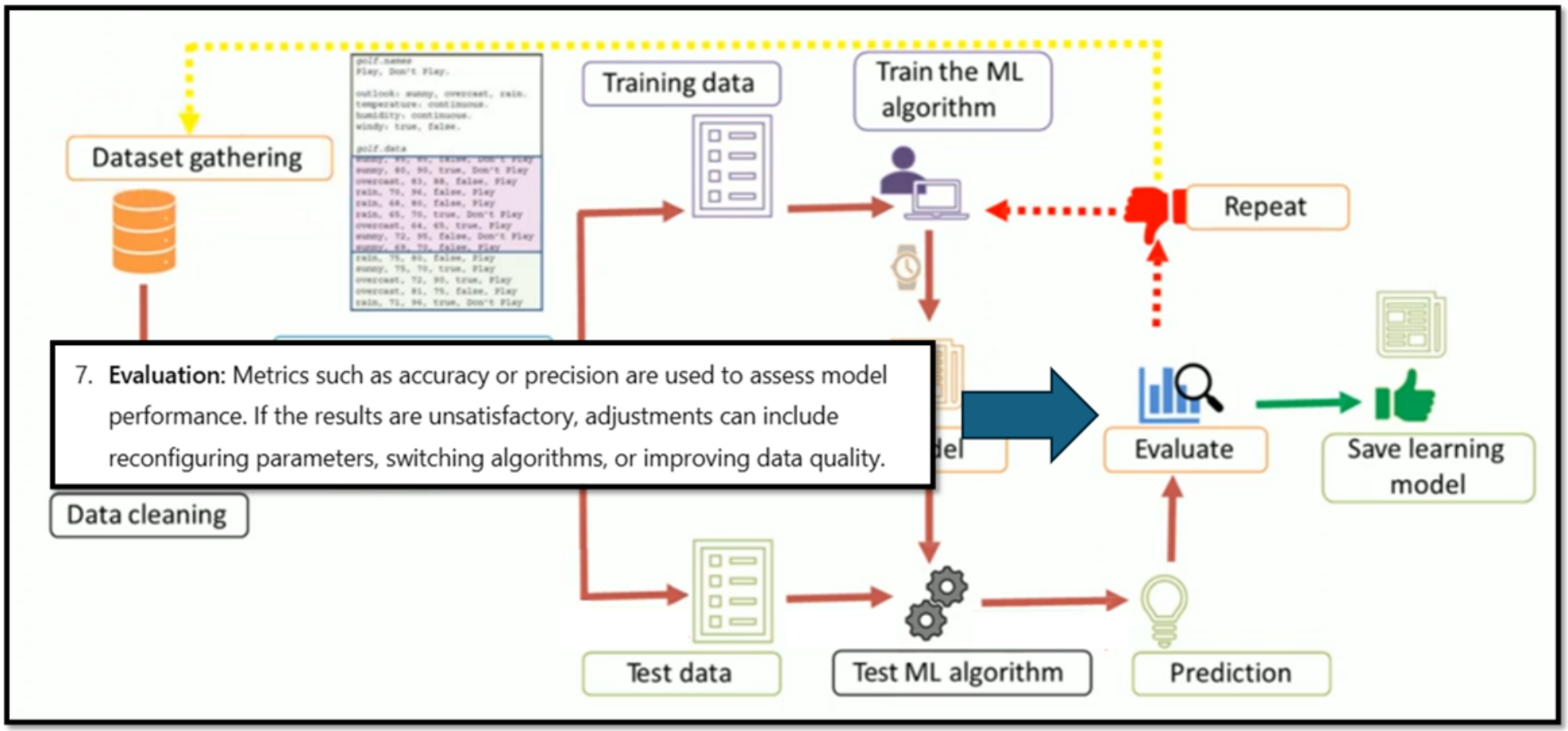
The Machine Learning Process

The development and deployment of ML models follow a structured, multi-step process:



The Machine Learning Process

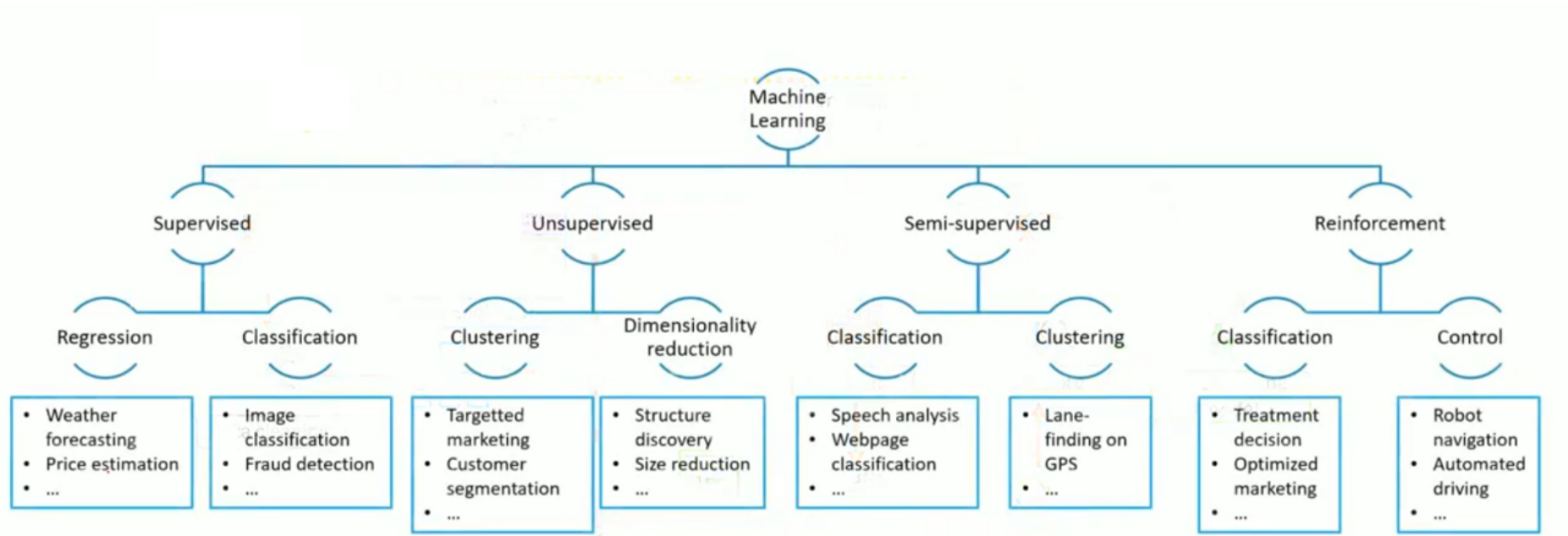
The development and deployment of ML models follow a structured, multi-step process:



Types of Machine Learning Models

Types of Machine Learning

ML algorithms can be broadly categorized into four types:



Supervised Learning

Types of Machine Learning

Supervised Learning

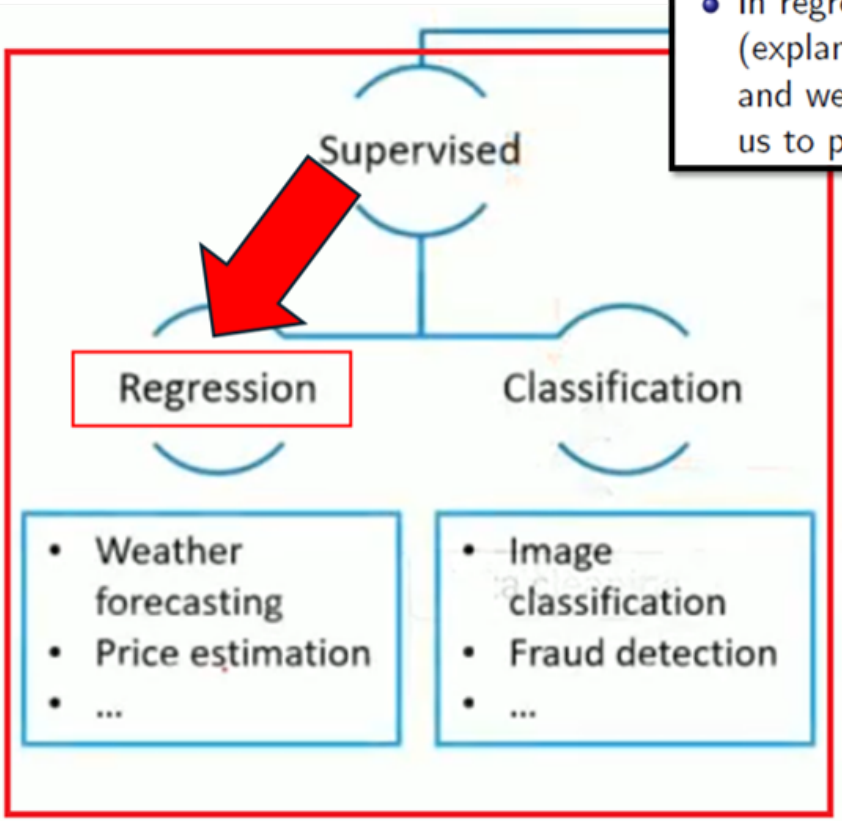
- In supervised learning, the algorithm is trained on labeled data, meaning both input variables (features) and corresponding output variables (labels) are provided.
- The goal is to learn the mapping between inputs and outputs.

Supervised learning includes:

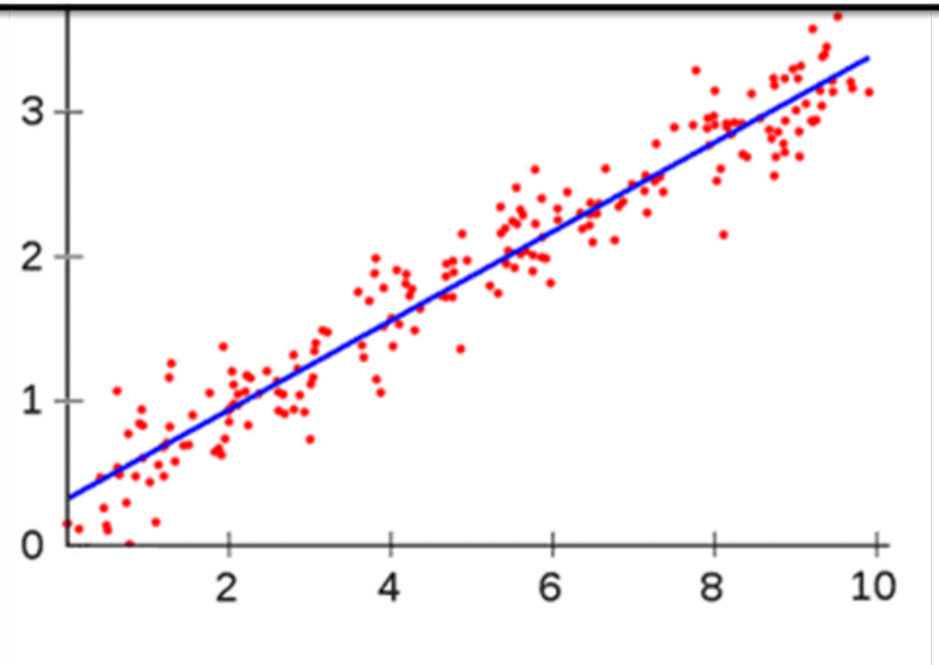
- **Regression**: Predicting continuous numerical values, such as temperatures or stock prices.
- **Classification**: Categorizing data into discrete groups, such as identifying whether an email is spam or not.

Supervised Learning

Types of Machine Learning

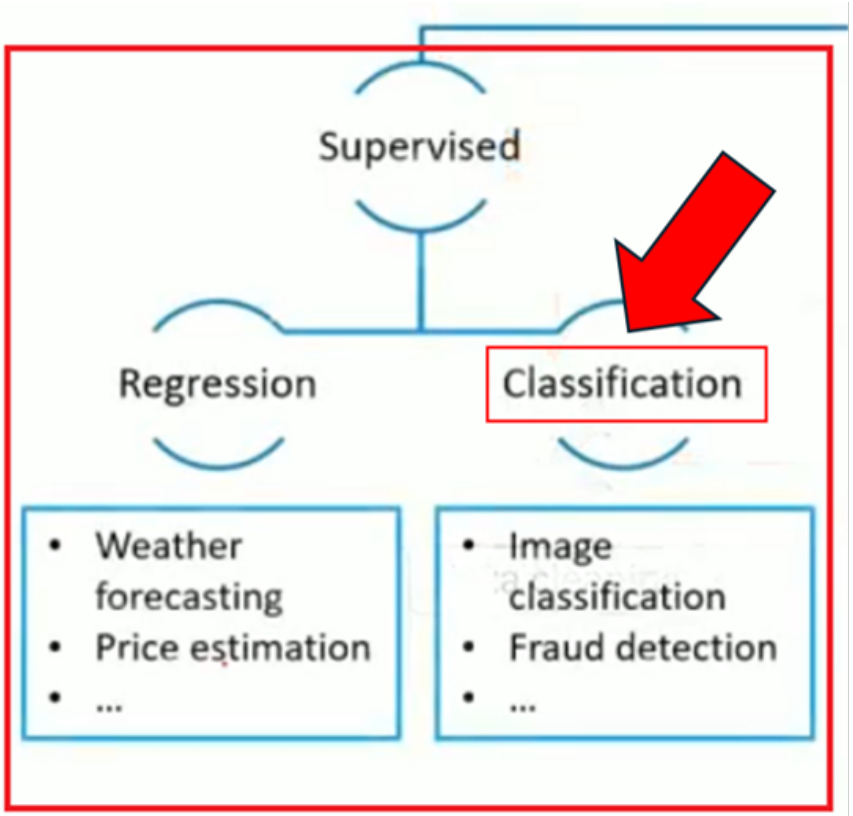


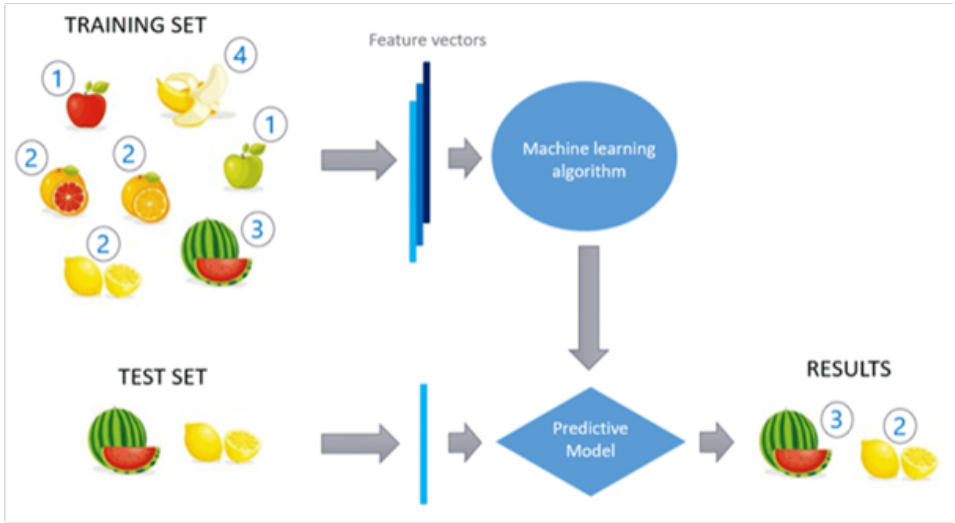
- In regression analysis, we are given a number of predictor (explanatory) variables and a continuous response variable (outcome), and we try to find a relationship between those variables that allows us to predict an outcome.



Supervised Learning

Types of Machine Learning





Supervised Learning

Types of Machine Learning

- From a formal poin of view, supervised learning process involves input variables, which we call X, and an output variable, which we call Y.
- We use an algorithm to learn the mapping function from the input to the output.
- In simple mathematics, the output Y is a dependent variable of input X as illustrated by:

$$Y = f(X)$$

Here, our end goal is to try to **approximate the mapping function** f, so that we can **predict** the output variables Y when we have new input data X.

Unsupervised Learning

Types of Machine Learning

Unsupervised Learning

- Unsupervised learning involves analyzing data without labeled outputs.
- The algorithm identifies hidden patterns or structures.

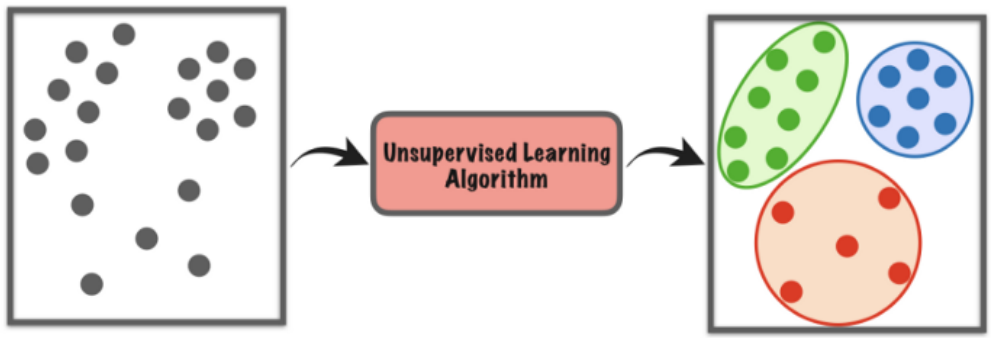
Two primary uses include:

- **Clustering**: Grouping similar data points, for instance, segmenting customers for targeted marketing.
- **Dimensionality Reduction**: Simplifying datasets by reducing features while preserving essential information, often used to speed up computations or eliminate noise.

Unsupervised Learning

Types of Machine Learning

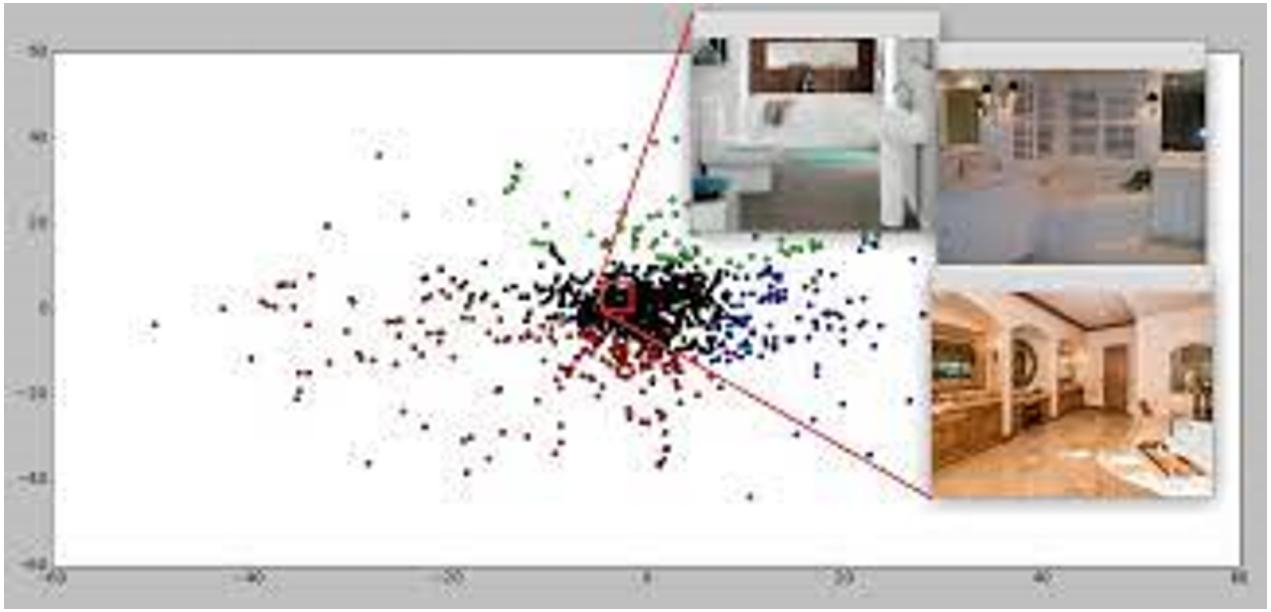
- Unsupervised algorithms attempt to find some sort of underlying structure in the data.
- Are some observations clustered into groups? Are there interesting relationships between different features? Which features carry most of the information?



Unsupervised Learning

Types of Machine Learning

- The data for **Unsupervised Learning** consists of features but no labels because the model is being used to identify patterns not to forecast something.
- For example we could use an unsupervised model to classify the houses that exist in a certain neghborhood without trying to predict any price.



Semi-Supervised Learning

Types of Machine Learning

Semi-Supervised Learning

- Semi-supervised learning combines elements of supervised and unsupervised methods.
- It operates on datasets where only some entries are labeled. Techniques may include assigning labels to unlabeled data or training the model iteratively on both labeled and unlabeled subsets.
- Applications include speech analysis and web page classification.

Reinforcement Learning

Types of Machine Learning

Reinforcement Learning

- Reinforcement learning involves algorithms that learn through interaction with their environment.
- These models improve by receiving rewards or penalties for their actions, refining their strategies over time.
- Applications range from training autonomous vehicles to optimizing marketing strategies.

Learning Tools

Subsection 1

Python and Anaconda

Python and Anaconda

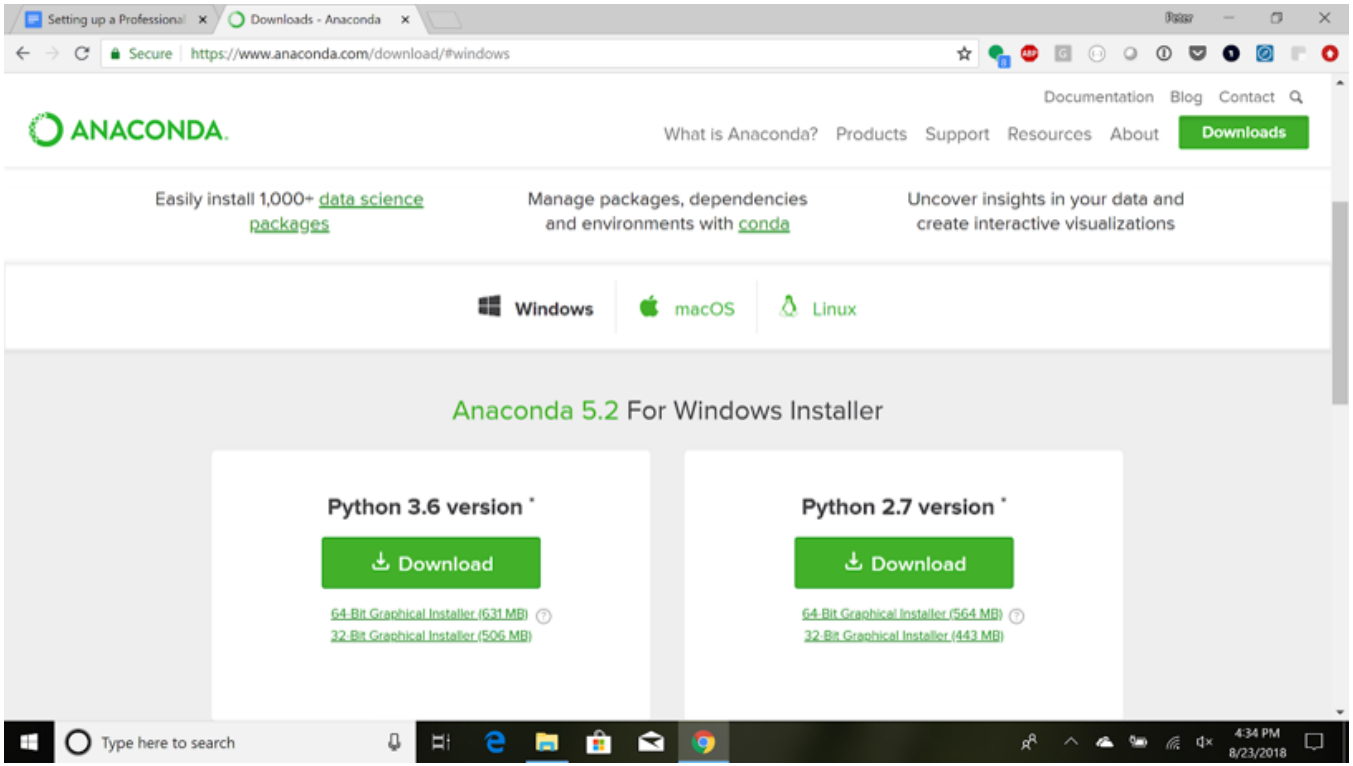
- Python is one of the most popular programming languages for data science and thanks to its very active developer and open source community, a large number of useful libraries for scientific computing and machine learning have been developed.
- Although the performance of interpreted languages, such as Python, for computation-intensive tasks is inferior to lower-level programming languages, extension libraries such as **NumPy**, **Matplotlib** and **Pandas**, among the others, have been developed that build upon lower-layer Fortran and C implementations for fast vectorized operations on multidimensional arrays.

Python and Anaconda

- For machine learning programming tasks, we will mostly refer to the **scikit-learn** library, which is currently one of the most popular and accessible open source machine learning libraries.
- In the last part of these lessons, when we focus on a subfield of machine learning called deep learning, we will use the latest version of the **Keras** library, which specializes in training so-called deep neural network models very efficiently.

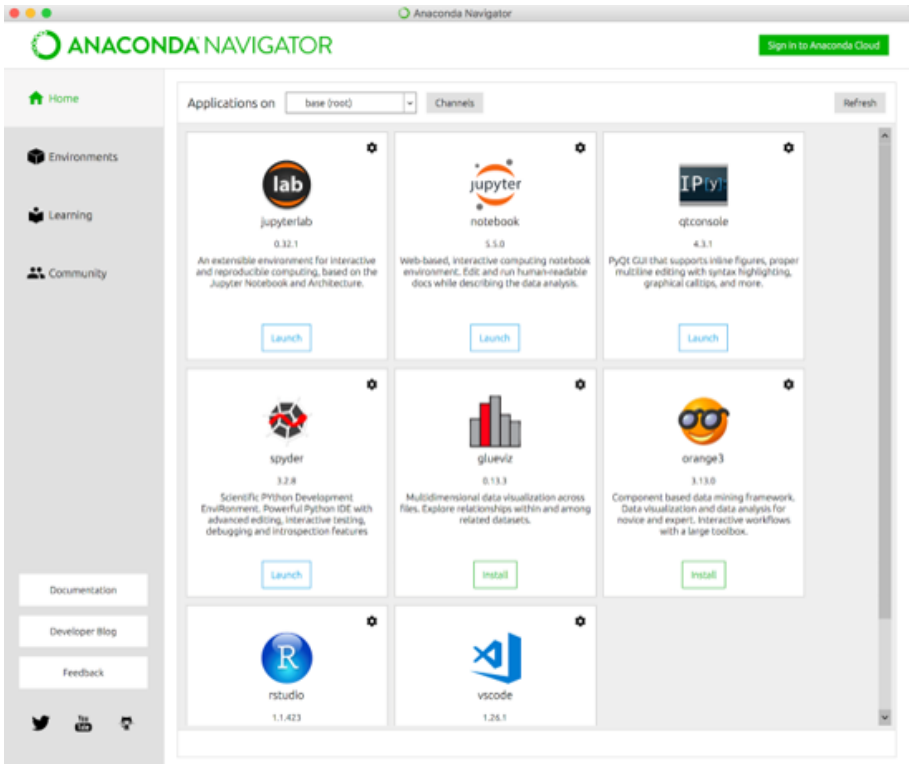
Installing Python and Anaconda

- To download an Anaconda distribution, you can use the official download page: **https://www.anaconda.com/download/**
- Here, you can select your platform and then choose the installer. For this, you can choose which version you want and whether 32-bit or 64-bit.



Testing Your Installation

To test your installation, on Windows, click on Start and then Anaconda Navigator in the program list (or search for Anaconda in the search bar and select Anaconda Navigator). On a Mac, open up the finder, and in the Applications folder, double click on Anaconda-Navigator.



Package Managers

- Anaconda will give you two package managers- pip and conda.
- When some packages aren't available with conda, you can use pip to install them.
- Note that using pip to install packages also available to conda may cause an installation error.
- After we have successfully installed Python, we can execute pip from the terminal to install additional Python packages:

pip install SomePackage

- Already installed packages can be updated via the –upgrade flag:

pip install SomePackage –upgrade

Subsection 2

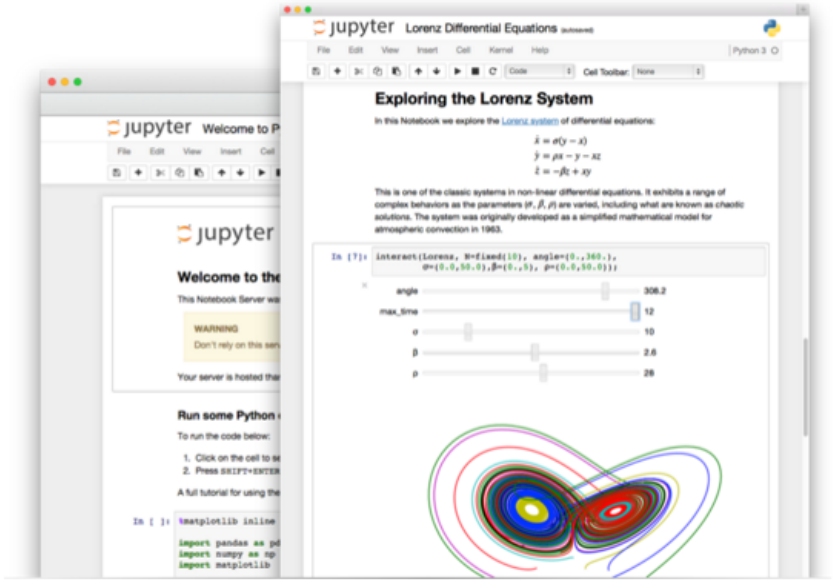
Jupyter Notebook

Teaching tools: Jupyter Notebook

- The Python world developed the IPython notebook system.
- Notebooks allow you to write text, but you insert code blocks as "cells" into the notebook.
- A notebook is interactive, so you can execute the code in the cell directly!
- Recently the Notebook idea took a much enhanced vision and scope, to explicitly allow languages other than Python to run inside the cells.
- Thus the Jupyter Notebook was born, a project initially aimed at Julia, Python and R (Ju-Pyt-e-R). But in reality many other languages are supported in Jupyter.

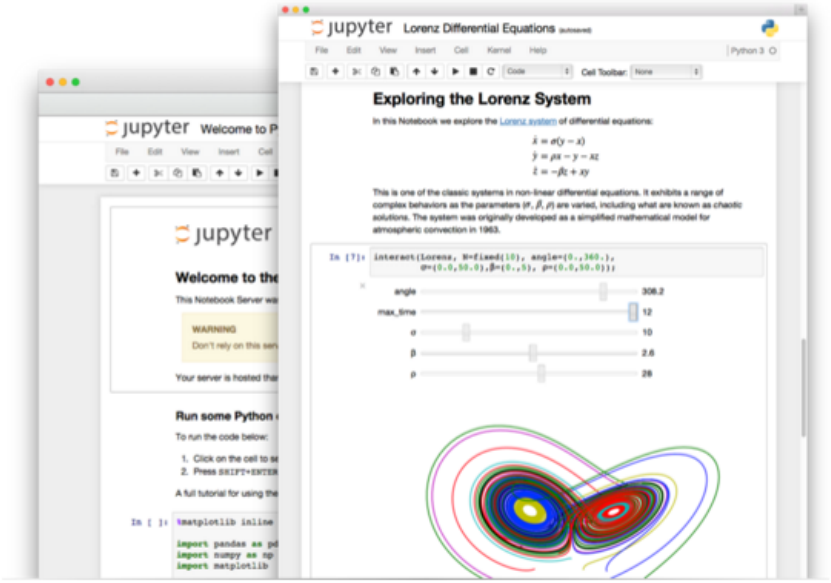
Teaching tools: Jupyter Notebook

- Jupyter was designed to enable sharing of notebooks with other people.
- The idea is that you can write some code, mix some text with the code, and publish this as a notebook.
- In the notebook they can see the code as well as the actual results of running the code.



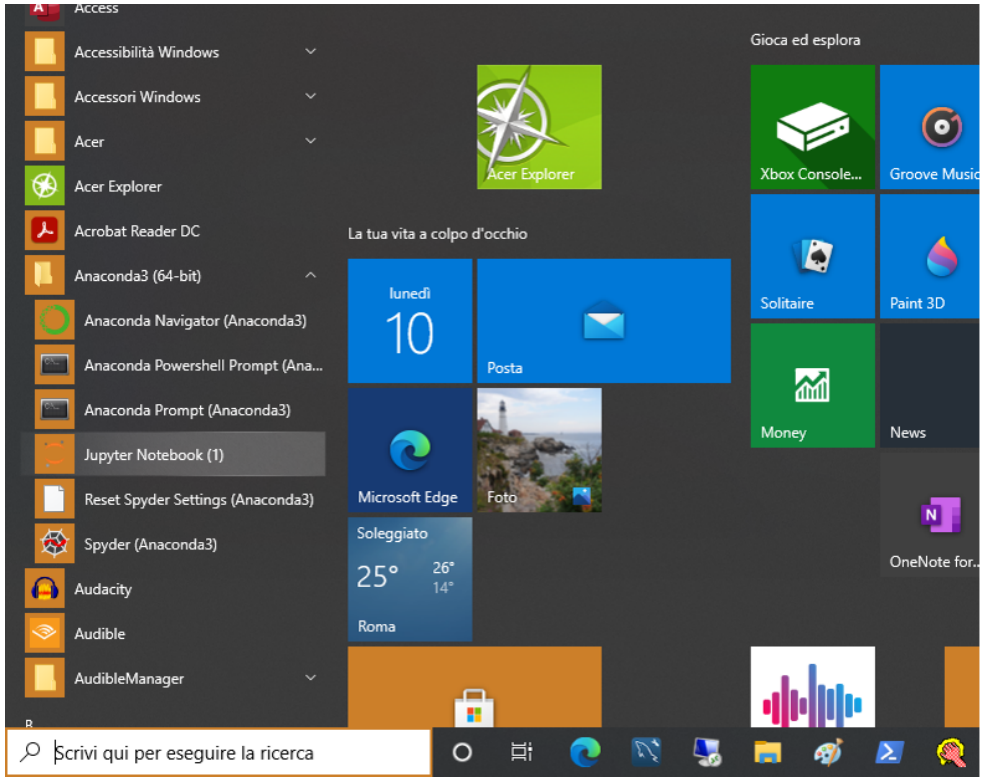
Teaching tools: Jupyter Notebook

- This is a nice way of sharing little experimental snippets, but also to publish more detailed reports with explanations and full code sets.
- Of course, a variety of web services allows you to post just code snippets (e.g. gist).
- What makes Jupyter different is that the service will actually render the code output.



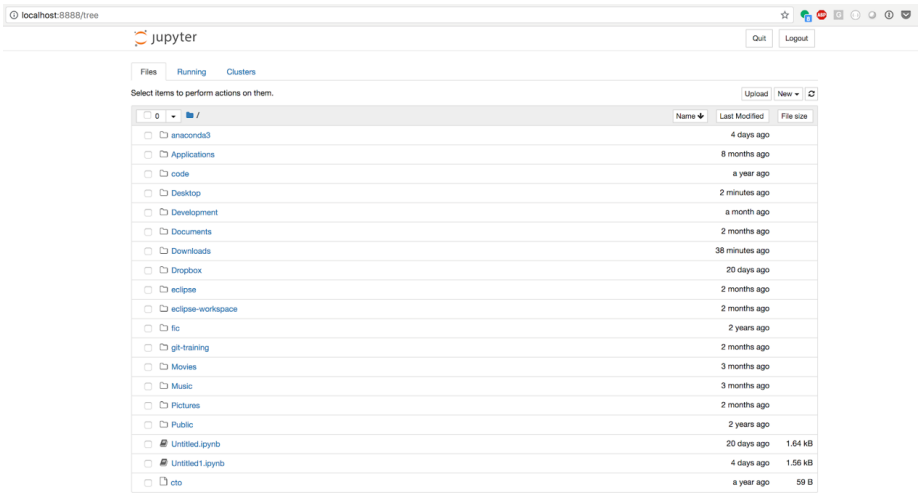
Teaching tools: Jupyter Notebook

- As we saw earlier, the Jupyter Notebook ships with Anaconda. To run it, you can get in your virtual environment and type the following command: **jupyter notebook**;
- Or directly from the Windows Menu...



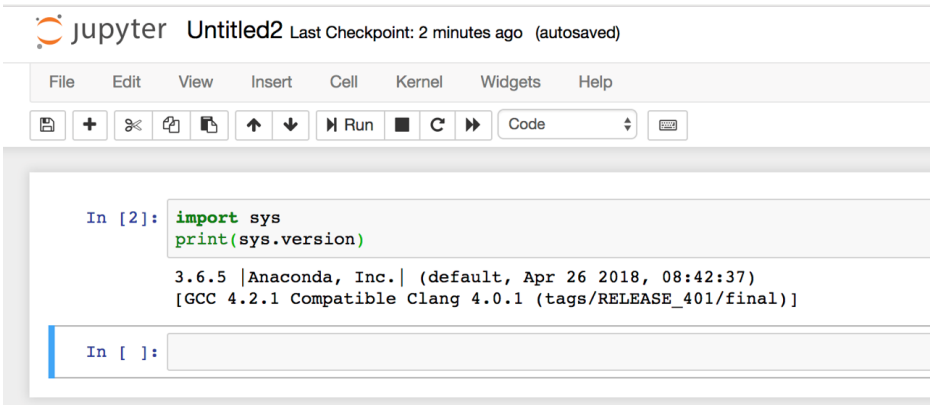
Teaching tools: Jupyter Notebook

- You can find this at **http://localhost:8888/tree**
- Now to run Python here, you can create a new file.



Teaching tools: Jupyter Notebook

To make sure it's working, click in the cell and type the following:



Subsection 3

Google Colab

Teaching tools: Google Colab

- Colaboratory, or "Colab" for short, is a product from Google Research.
- Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

https://colab.research.google.com/notebooks/intro.ipynb?hl=en

Teaching tools: Google Colab

- More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.
- Colab notebooks are stored in *Google Drive*, or can be loaded from *GitHub*. Colab notebooks can be shared just as you would with Google Docs or Sheets.

https://colab.research.google.com/notebooks/intro.ipynb?hl=en