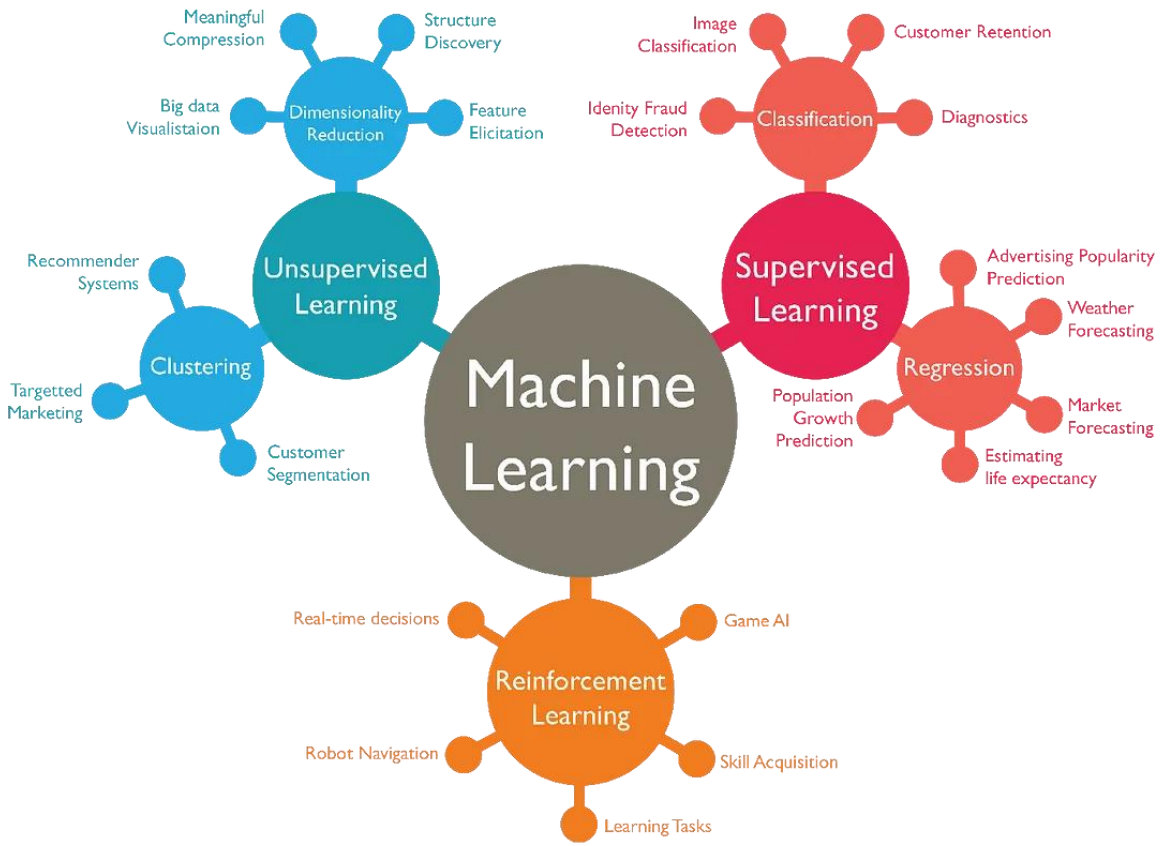
Machine Learning & Model Development



Source: https://www.geeksforgeeks.org/steps-to-build-a-machine-learning-model/

https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b

Understanding Machine Learning

We can see machine learning as a subset or just a part of artificial intelligence that focuses on developing algorithms that are capable of learning hidden patterns and relationships within the data allowing algorithms to generalize and make better predictions or decisions on new data.

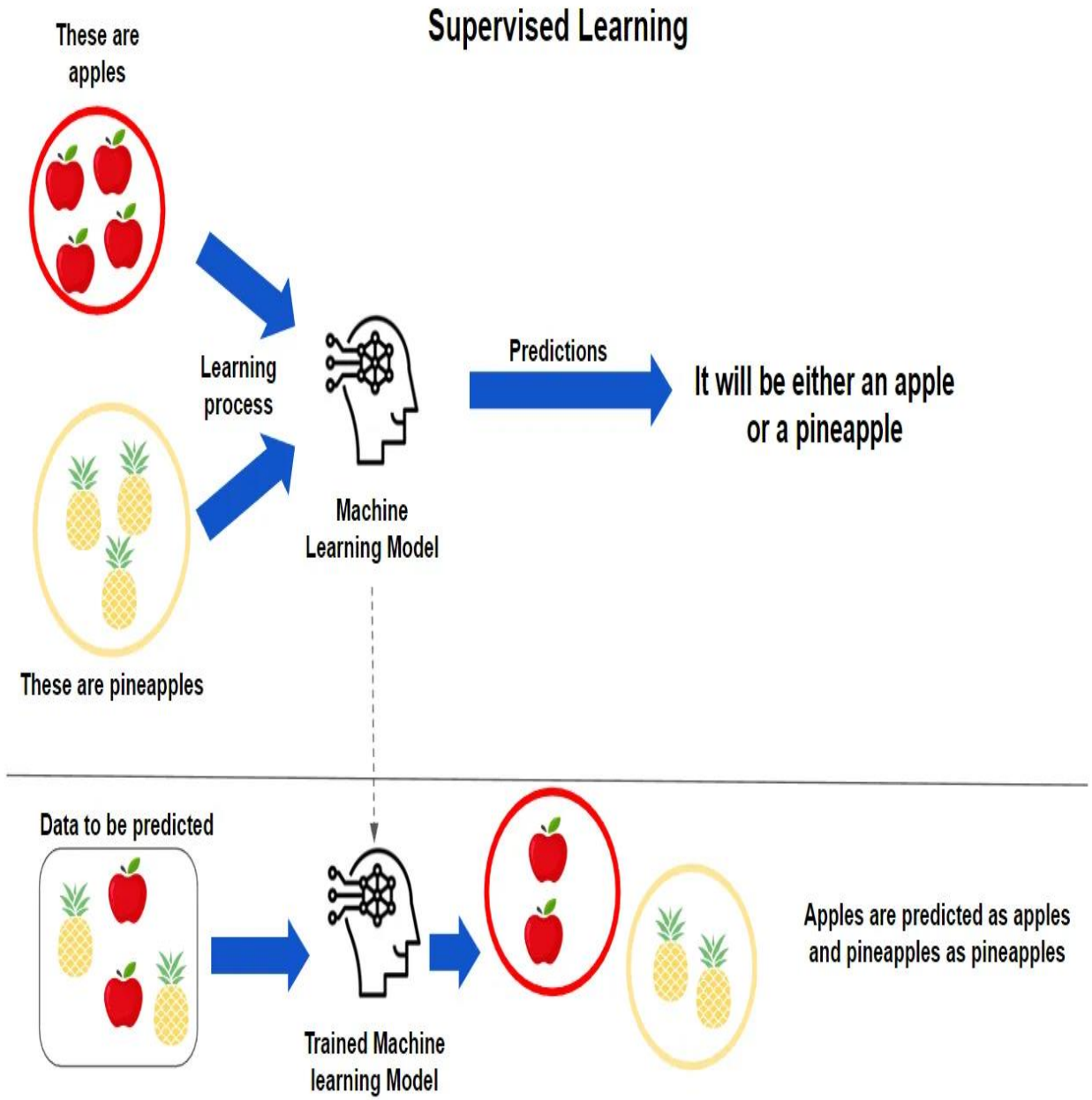
To achieve this, we have to understand several key concepts and techniques like

- supervised learning,
- unsupervised learning, and
- reinforcement learning.

Source: https://www.geeksforgeeks.org/steps-to-build-a-machine-learning-model/

Supervised Learning

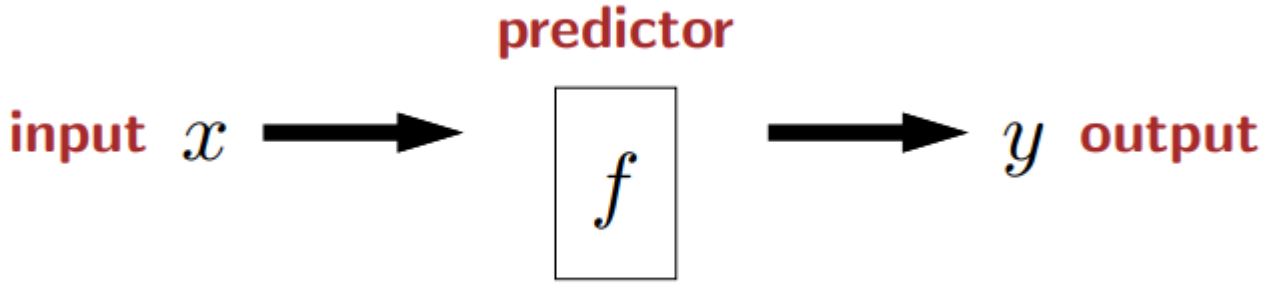
- **Supervised learning** involves training a model on labeled data, where the algorithm learns from the input data and its corresponding target (output labels).
- The goal is to map from input to output, allowing the model to learn the relationship and make predictions based on the learnings of new data.
- Some of its algorithms are linear regression, logistic regression decision trees, and more.



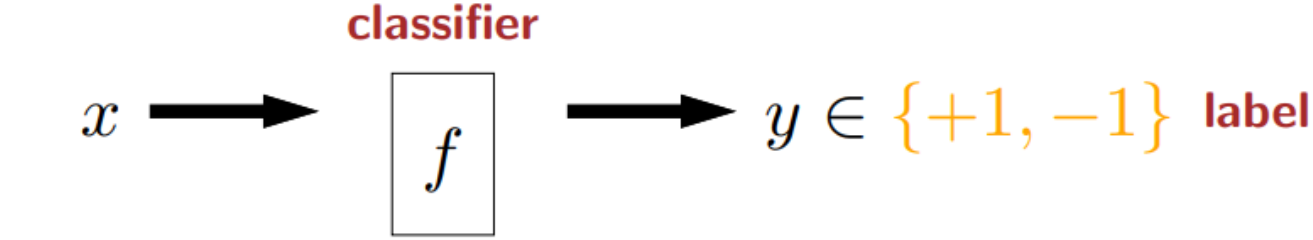
https://medium.datadriveninvestor.com/exploring-the-basics-of-supervised-learning-bfc18fe5ce72

Supervised Learning

- After training, learning model (the supervised machine learning algorithm, predictor) ***f*** takes some input vector ***x*** and produces some output ***y***.
- The input can usually be arbitrary (an image, a vector of values or sentence), but the form of the output y is generally restricted, and what it is determines the type of prediction task.



Supervised Learning : Classification



Fraud detection: credit card transaction $\rightarrow$ fraud or no fraud

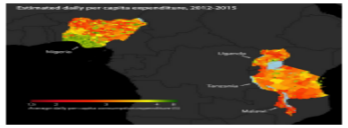
Toxic comments: online comment $\rightarrow$ toxic or not toxic

- One common prediction task is binary classification, where the output y, typically expressed as positive (+1) or negative (-1).
- In the context of classification tasks, f is called a classifier and y is called a label (sometimes class, category, or tag).
- Multiclass classification is a generalization of binary classification where the output y could be one of K possible values. For example, in digit classification, K =10.

Source : CS221

Supervised Learning : Regression

$$x \longrightarrow \boxed{f} \longrightarrow y \in \mathbb{R} \text{ response}$$



Poverty mapping: satellite image $\longrightarrow$ asset wealth index

Housing: information about house $\longrightarrow$ price

Arrival times: destination, weather, time $\longrightarrow$ time of arrival

- The second major type of prediction task is regression. Here, the output y is a real number (often called the response or target).
- The key distinction between classification and regression is that classification has discrete outputs (e.g., ”yes” or ”no” for binary classification), whereas regression has continuous outputs.

Supervised Learning : Structured prediction

$x \longrightarrow \boxed{f} \longrightarrow y$ is a complex object

Machine translation: English sentence → Japanese sentence

Dialogue: conversational history → next utterance

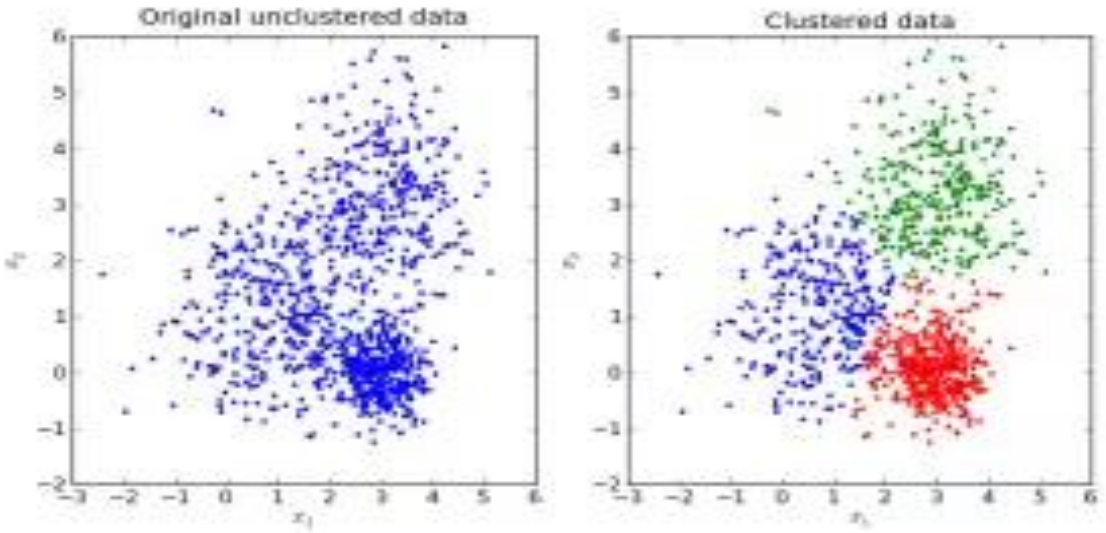
Image captioning: image → sentence describing image

Image segmentation: image → segmentation

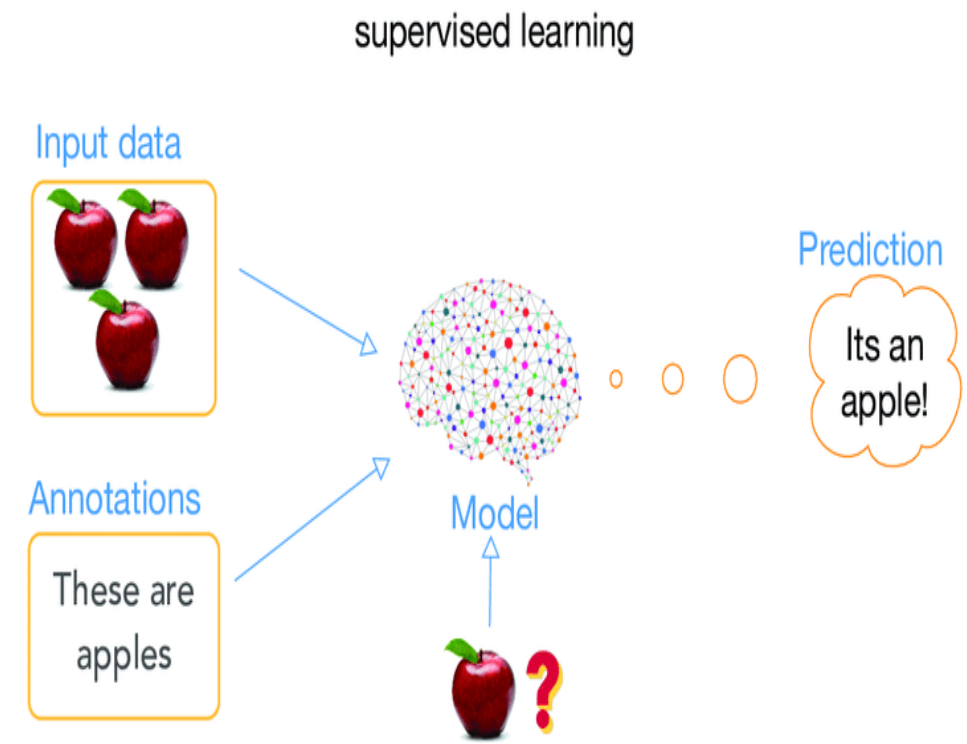
- In structured prediction, the output y is a complex object, which could be a sentence or an image.

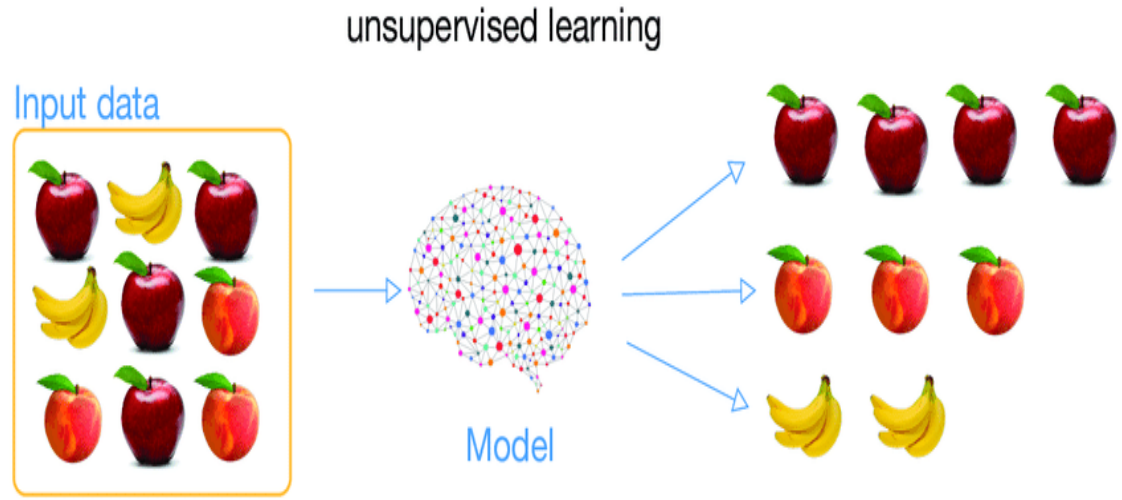
Unsupervised learning

- **Unsupervised learning**, on the other hand, deals with the unlabeled dataset where algorithms try to uncover hidden patterns or structures within the data.
- Unlike **supervised learning** which depends on labeled data to create patterns or relationships for further predictions, unsupervised learning operates without such guidance.
- Some of its algorithms are, Clustering algorithms like k-means, hierarchical clustering dimensionality reduction algorithms like PCA, and more.



https://www.linedata.com/what-unsupervised-machine-learning

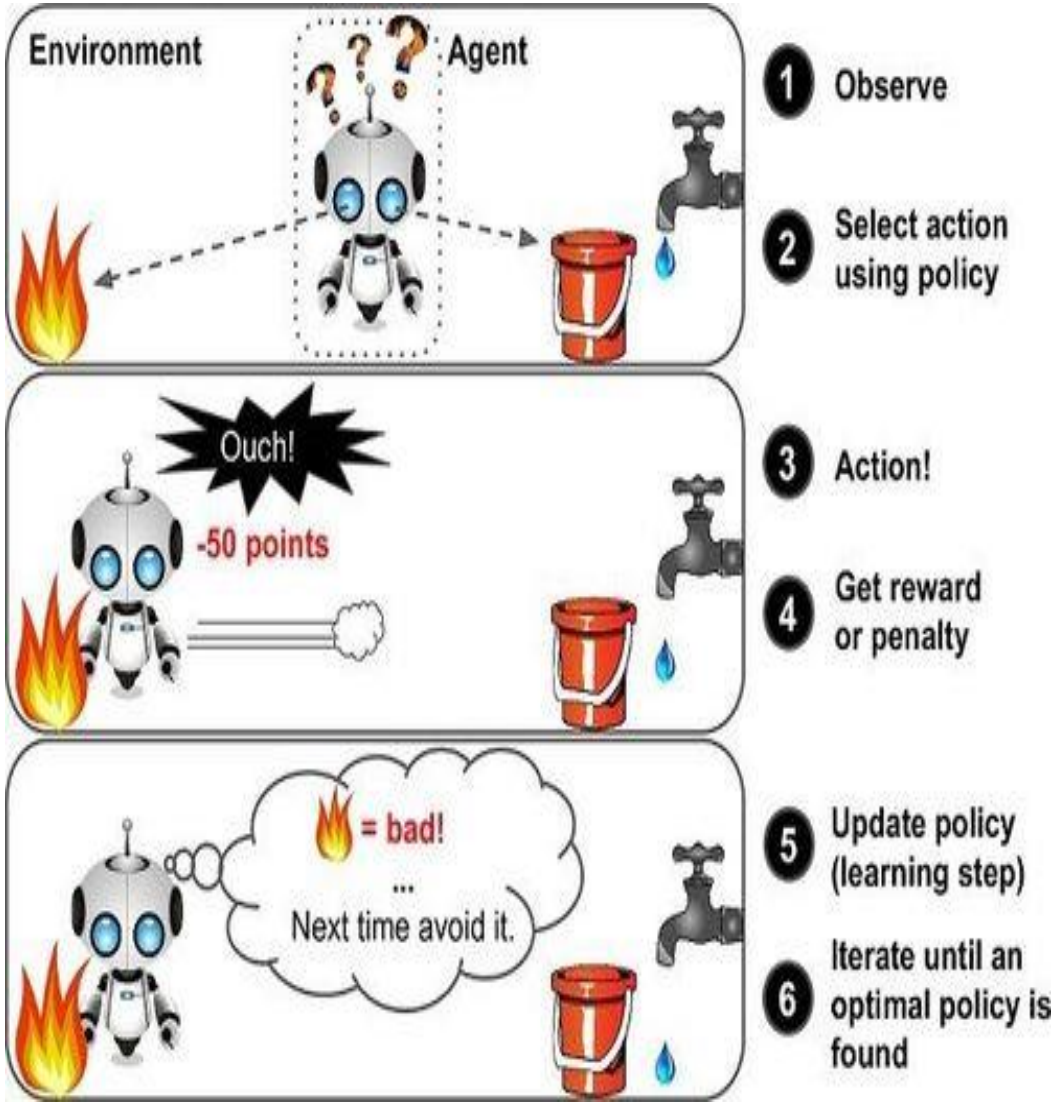




https://nadeemm.medium.com/supervised-vs-unsupervised-learning-8b559f992152

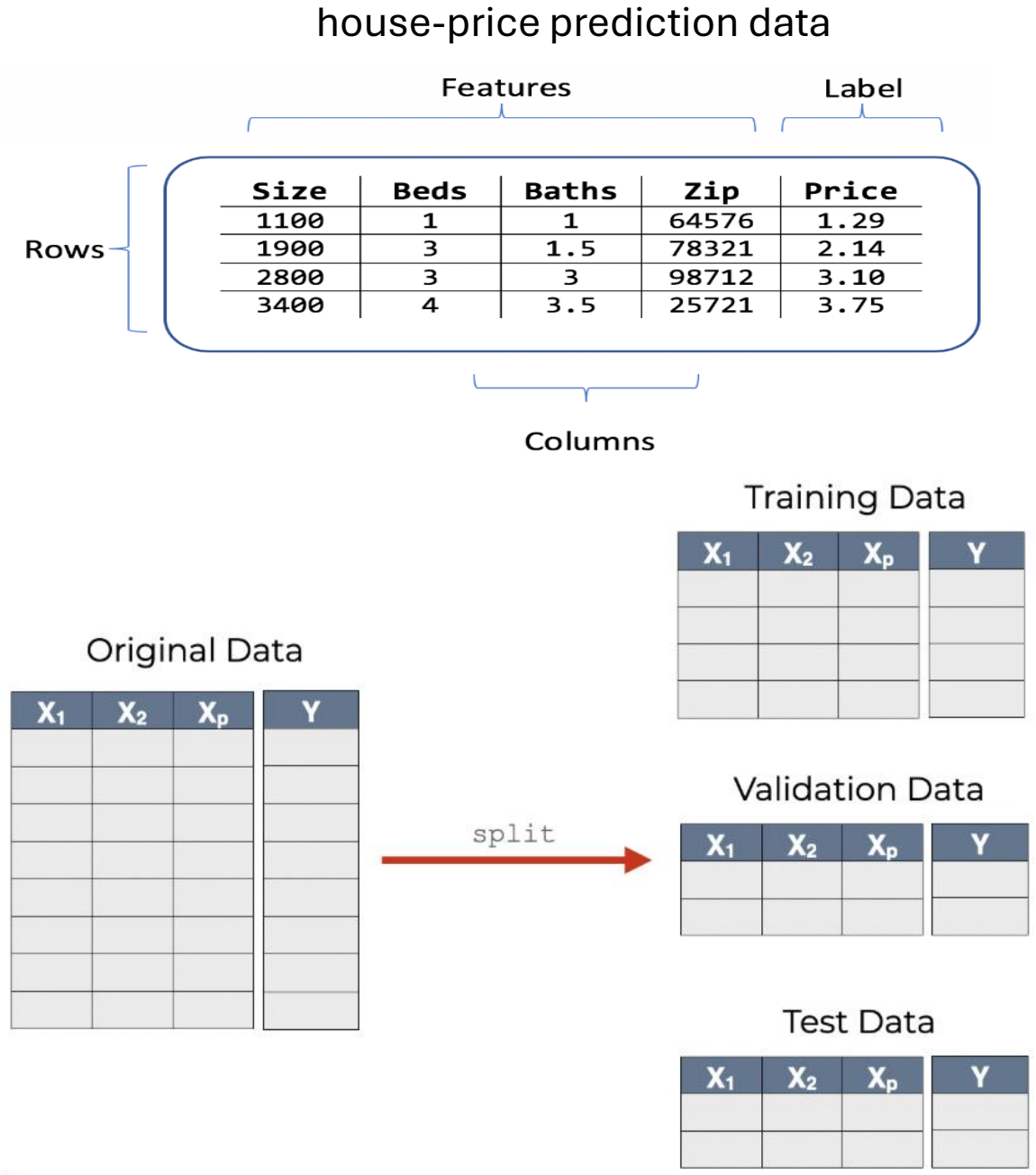
Reinforcement learning

- **Reinforcement learning** is a part of machine learning that involves training an agent to interact with an environment and learn optimal actions through trial and error.
- It employs a reward-penalty strategy, the agent receives feedback in the form of rewards or penalties based on its actions, allowing it to learn from experience and maximize its reward over time.
- Reinforcement learning applications in areas such as robotics, games, and more.



Key Machine Learning Terminologies

1. **Features:** These are the input variables or attributes used by the model to make predictions.
2. **Labels:** The output or target variable that the model predicts in supervised learning.
3. **Training Set:** A subset of the data used to train the model by identifying patterns.
4. **Validation Set:** Data used to tune the model's hyperparameters and optimize performance.
5. **Test Set:** Unseen data used to evaluate the model's final performance.



Steps to Build a Machine Learning Model

Step 1: Data Collection for Machine Learning

Step 2: Data Preprocessing and Cleaning

Step 3: Selecting the Right Machine Learning Model

Step 4: Training Your Machine Learning Model

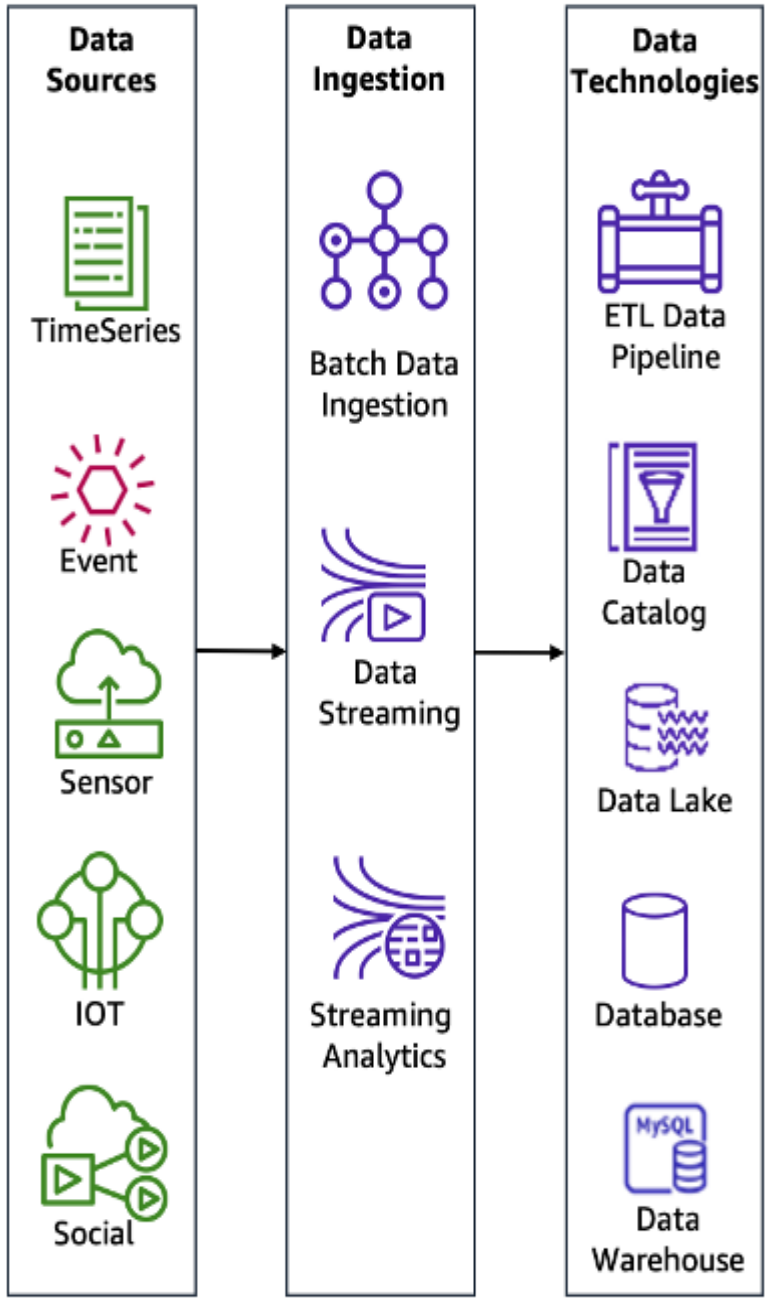
Step 5: Evaluating Model Performance

Step 6: Tuning and Optimizing Your Model

Step 7: Deploying the Model and Making Predictions

Step 1: Data Collection for Machine Learning

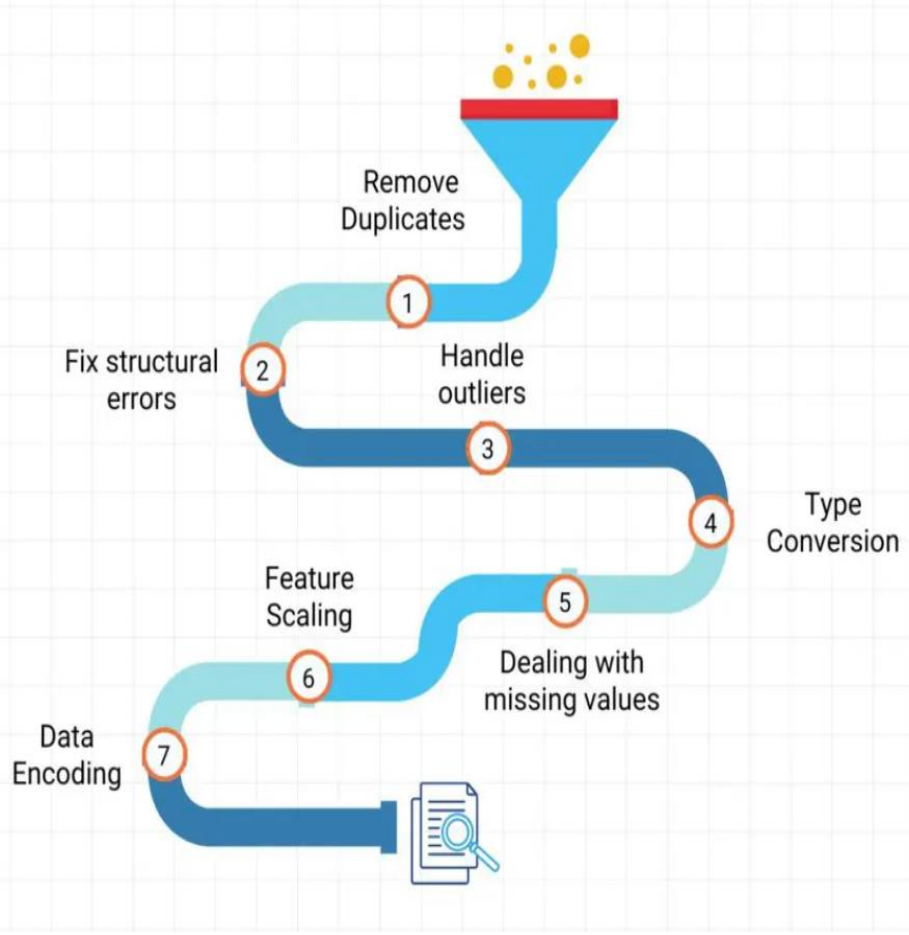
- Relevant data is gathered from various sources to train the machine learning model and enable it to make accurate predictions.
- The first step in data collection is defining the problem and understanding the requirements of the machine learning project.
- This usually involves determining the type of data we need for our project like structured or unstructured data, and identifying potential sources for gathering data.
- Once the requirements are finalized, data can be collected from a variety of sources such as databases, APIs, web scraping, and manual data entry.
- It is crucial to ensure that the collected data is both relevant and accurate, as the quality of the data directly impacts the generalization ability of our machine learning model.
- In other words, the better the quality of the data, the better the performance and reliability of our model in making predictions or decisions.



https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/data-collection.html

Step 2: Data Preprocessing and Cleaning

- Preprocessing and preparing data is an important step that involves transforming raw data into a format that is suitable for training and testing for our models. This phase aims to clean i.e. remove null values, and garbage values, and normalize and preprocess the data to achieve greater accuracy and performance of our machine learning models.
- The preprocessing process typically involves several steps, including handling missing values, encoding categorical variables i.e. converting into numerical, scaling numerical features, and feature engineering.
- This ensures that the model's performance is optimized and also our model can generalize well to unseen data and finally get accurate predictions.



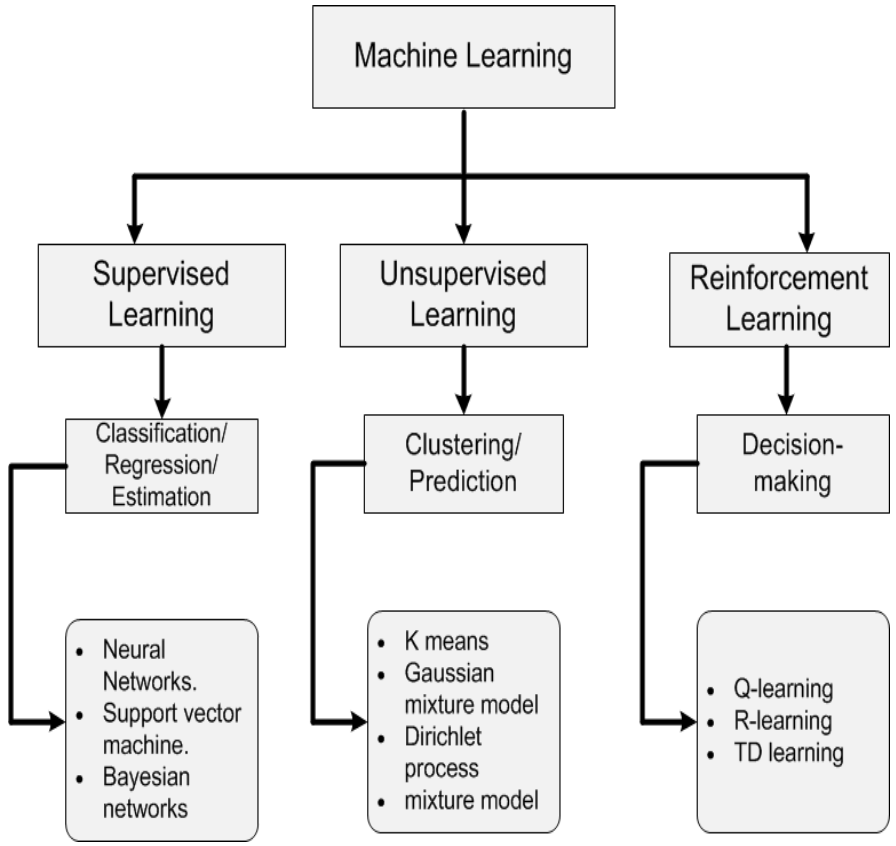
Step 3: Selecting the Right Machine Learning Model

- Selecting the right **machine learning model** plays a pivotal role in building of successful model, with the presence of numerous algorithms and techniques available easily, choosing the most suitable model for a given problem significantly impacts the accuracy and performance of the model.

- The process of selecting the right machine learning model involves several considerations, some of which are:

  - Firstly, understanding the nature of the problem is an essential step, as our model nature can be of any type like classification, regression, clustering or more, different types of problems require different algorithms to make a predictive model.

  - Secondly, familiarizing yourself with a variety of machine learning algorithms suitable for your problem type is crucial. Evaluate the complexity of each algorithm and its interpretability. We can also explore more complex models like deep learning may help in increasing your model performance but are complex to interpret.

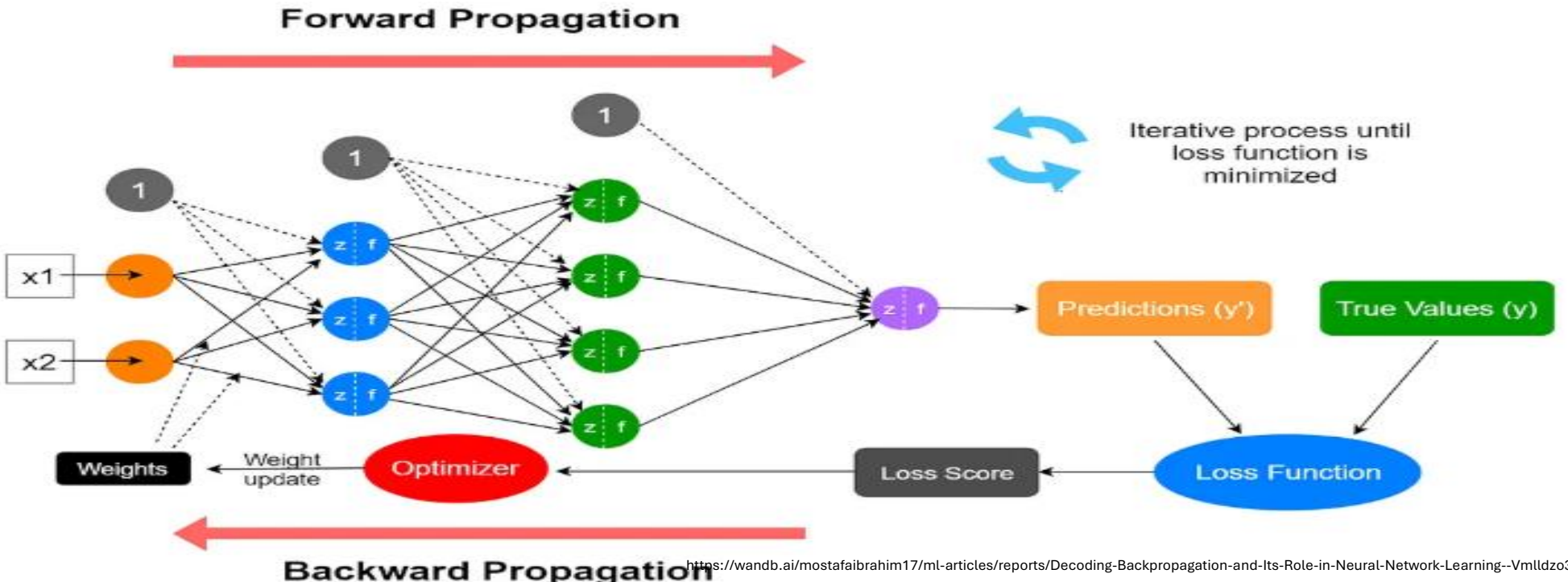


Step 4: Training Your Machine Learning Model

- Use prepared training data to teach the model to recognize patterns and make predictions based on the input features.
- During the training process, we begin by feeding the preprocessed data into the selected machine-learning algorithm.
- The algorithm then iteratively adjusts its internal parameters (weights) to minimize the difference between its predictions and the actual target values in the training data. This optimization process often employs techniques like gradient descent.
- As the model learns from the training data, it gradually improves its ability to generalize to new or unseen data. This iterative learning process enables the model to become more adept at making accurate predictions across a wide range of scenarios.

Training Data

x1	x2	y
1	3	0
1	7	1
3	6	0



https://wandb.ai/mostafaibrahim17/ml-articles/reports/Decoding-Backpropagation-and-Its-Role-in-Neural-Network-Learning--Vmlldzo3MTY5MzM1

Step 5: Evaluating Model Performance

- Once you have trained your model, it's time to assess its performance. There are various metrics used to evaluate model performance, categorized based on the type of task: regression/numerical or classification.

- **For regression tasks, common evaluation metrics are:**

  - **Mean Absolute Error (MAE):** MAE is the average of the absolute differences between predicted and actual values.

  - **Mean Squared Error (MSE):** MSE is the average of the squared differences between predicted and actual values.

  - **Root Mean Squared Error (RMSE):** It is a square root of the MSE, providing a measure of the average magnitude of error.

  - **R-squared (R2):** It is the proportion of the variance in the dependent variable that is predictable from the independent variables.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

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

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

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ – *predicted value of y*

$\bar{y}$ – *mean value of y*

Step 5: Evaluating Model Performance

- **For classification tasks, common evaluation metrics are:**
  - **Accuracy:** Proportion of correctly classified instances out of the total instances.
  - **Precision:** Proportion of true positive predictions among all positive predictions.
  - **Recall:** Proportion of true positive predictions among all actual positive instances.

EXAMPLE

Suppose we have a medical test to detect a rare disease, and the model's predictions are as follows:

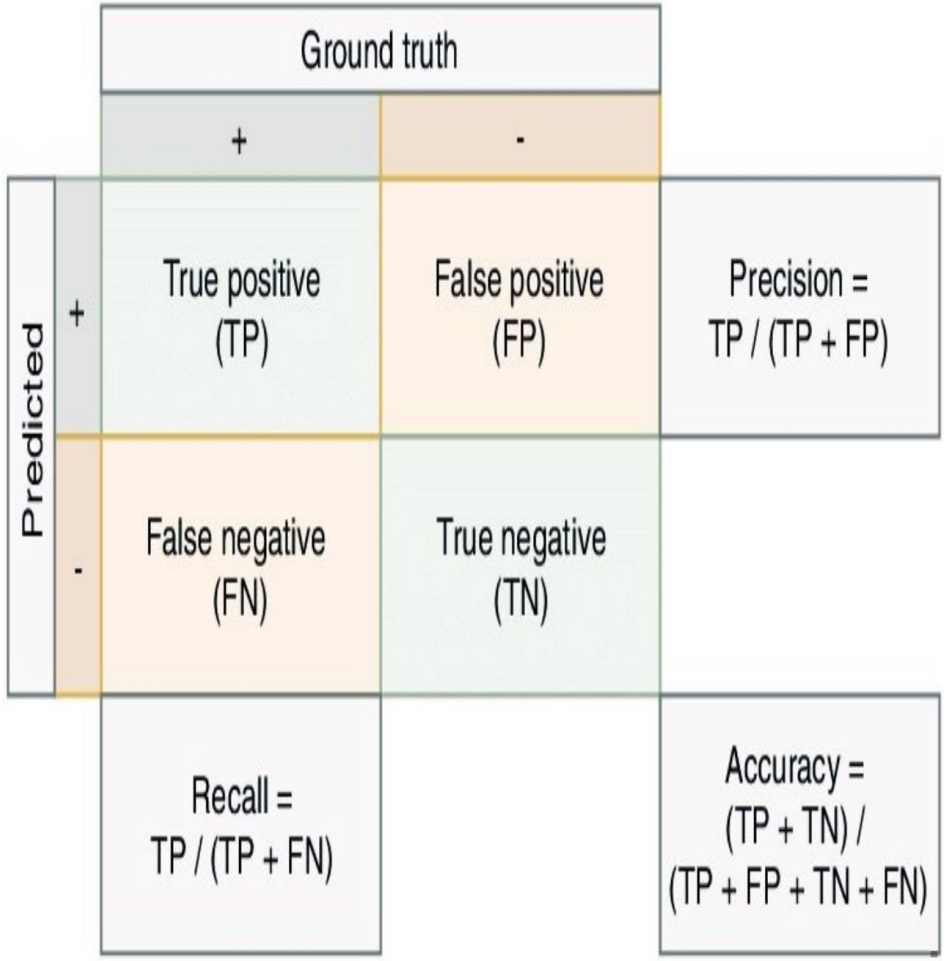
- True Positives (TP) = 40 (correctly identified diseased patients)
- False Positives (FP) = 10 (healthy people wrongly diagnosed with the disease)
- False Negatives (FN) = 20 (diseased patients missed by the model)
- True Negatives (TN) = 1000 (correctly identified healty patients

	+	-
+	40	10
-	20	930

ACCURACY = (40+930) / (40+10+20+030= 970/1000=0,97

PRECISION = 40 / (40+10) = 0.80

RECALL = 40 / (40+20) = 0,66

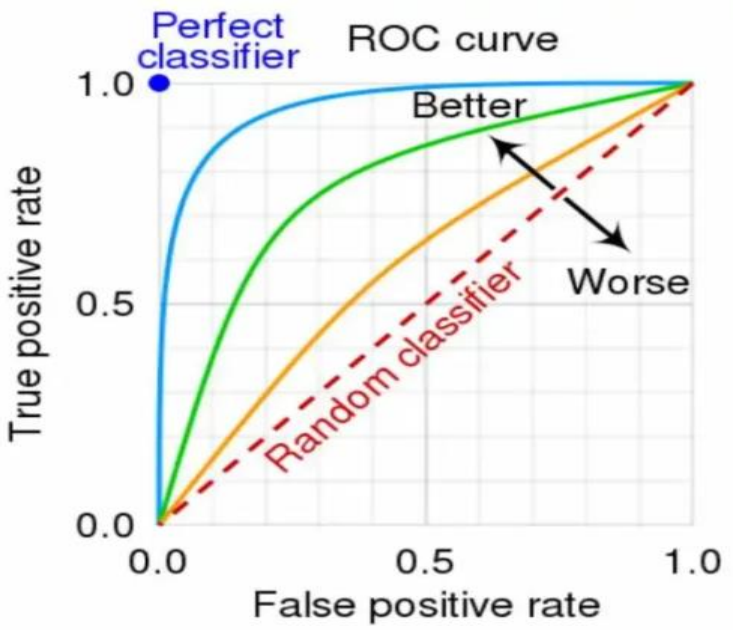


FORMULAS

Metric	Formula	Description
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall/Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1-score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of Precision and Recall

Step 5: Evaluating Model Performance

- **For classification tasks, common evaluation metrics are:.**
  - **Area Under the Receiver Operating Characteristic curve (AUC-ROC):** Measure of the model's ability to distinguish between classes.
  - **Confusion Metrics:** It is a matrix that summarizes the performance of a classification model, showing counts of true positives, true negatives, false positives, and false negatives instances.



		Predicted: Dog	Predicted: Cat	
Actual	Dog	24	6	30
	Cat	2	18	20

		Predicted: On time	Predicted: Late	Predicted: Very late	
Actual	On time	40	7	3	50
	Late	5	25	5	35
	Very late	3	5	8	15

Step 6: Tuning and Optimizing Your Model

As we have trained our model, our next step is to optimize our model more. Tuning and optimizing helps our model to maximize its performance and generalization ability.

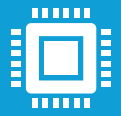
This process involves fine-tuning hyperparameters, selecting the best algorithm, and improving features through feature engineering techniques.

Hyperparameters are parameters that are set before the training process begins and control the behavior of the machine learning model. These are like learning rate, regularization and parameters of the model should be carefully adjusted.

Step 7: Deploying the Model and Making Predictions

Deploying the model and making predictions is the final stage in the journey of creating an ML model. Once a model has been trained and optimized, it's to integrate it into a production environment where it can provide real-time predictions on new data.

During model deployment, it's essential to ensure that the system can handle high user loads, operate smoothly without crashes, and be easily updated.



Tools like Docker and Kubernetes help make this process easier by packaging the model in a way that makes it easy to run on different computers and manage efficiently.

Once deployment is done our model is ready to predict new data, which involves feeding unseen data into the deployed model to enable real-time decision making.