# 6COSC020W: APPLIED AI

# WEEK 7: MACHINE LEARNING (ML)

DR. ESTER BONMATI

(e.bonmaticoll@westminster.ac.uk)

# NOTES

- We will use Poll Everywhere multiple times during the lecture. Please have your device ready.

# LEARNING OUTCOMES OF THIS SESSION

**Machine Learning**

- **Supervised**
- **Unsupervised**
- Reinforcement Learning

**Supervised**

- Regression vs. Classification
- **Linear regression**
- **Decision trees**
- Applications
- Research

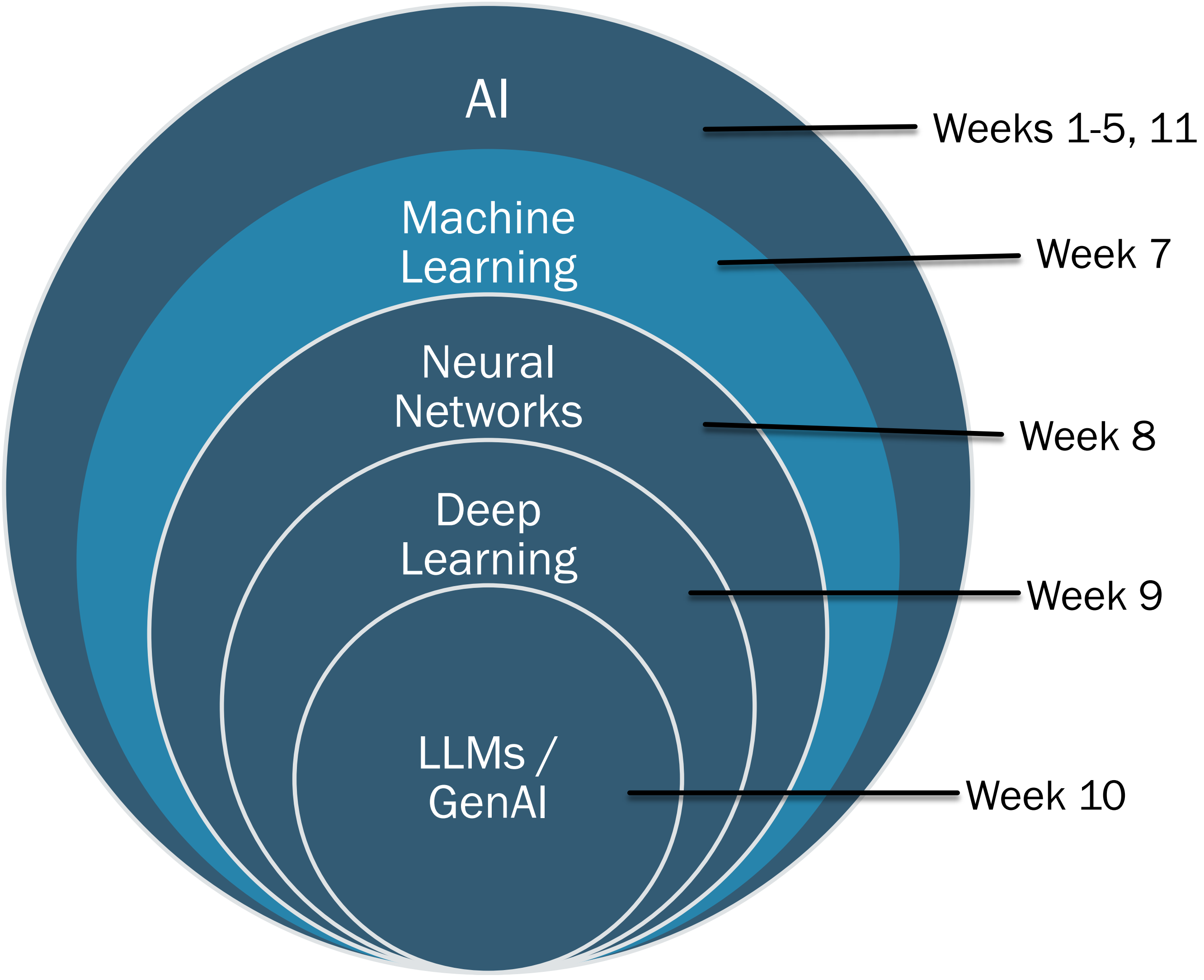
**Unsupervised**

- **K-means**
- Applications
- Research

**Data**

- Numerical vs. categorical
- **Data split**
- **Accuracy metrics**
- Confusion matrix

# NEURAL NETWORKS AND ARTIFICIAL INTELLIGENCE



# What do you think is machine learning?

Nobody has responded yet.

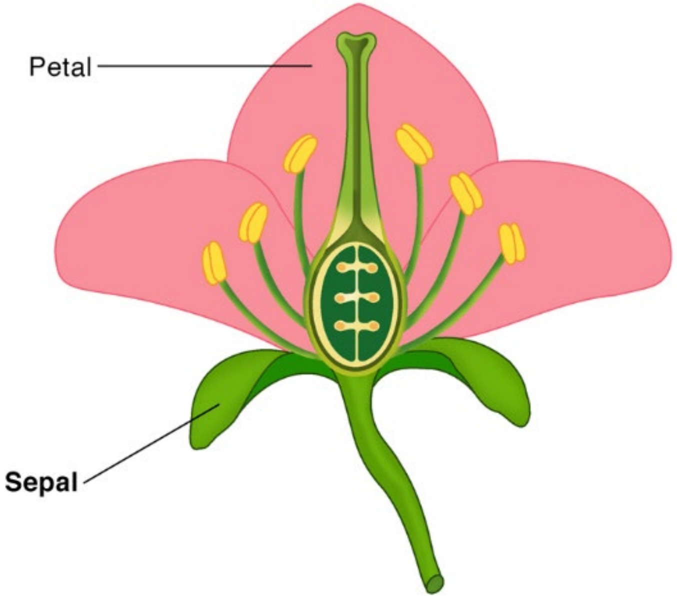
Hang tight! Responses are coming in.


LET’S START WITH
A PROBLEM…

# MACHINE LEARNING: EXAMPLE

We have the following dataset*:

| sepal_length | sepal_width | species |
|---|---|---|
| 5.1 | 3.5 | setosa |
| 4.9 | 3.0 | setosa |
| 4.7 | 3.2 | setosa |
| 4.6 | 3.1 | setosa |
| 5.0 | 3.6 | setosa |
| 6.7 | 3.0 | virginica |
| 6.3 | 2.5 | virginica |
| 6.5 | 3.0 | virginica |
| 6.2 | 3.4 | virginica |
| 5.9 | 3.0 | virginica |




We have a new set of measurements:
Sepal length = **6.4**
Sepal width = **3.1**

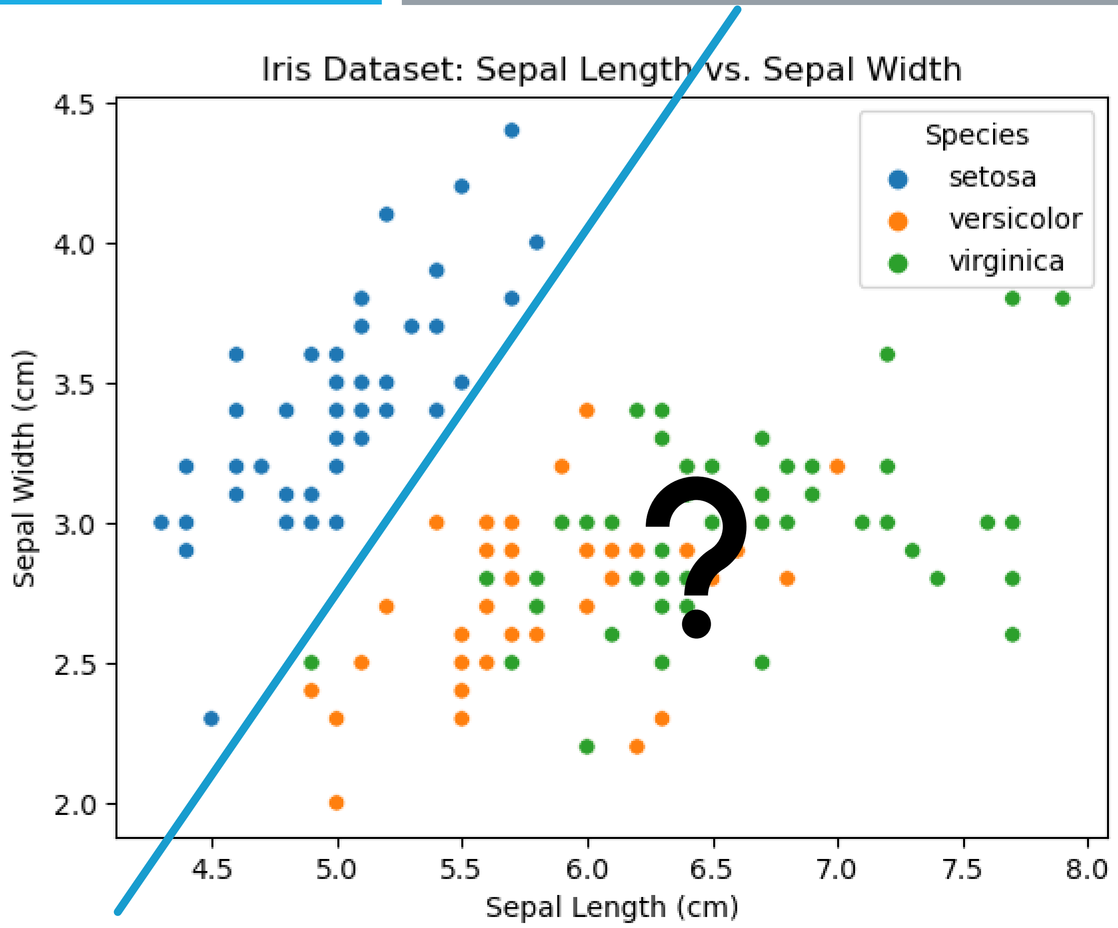
**Which species is it?**

*Extracted from the iris dataset

# MACHINE LEARNING: EXAMPLE

We add a new species (virginica)

| | sepal_length | sepal_width | species |
|---|---|---|---|
| 0 | 5.1 | 3.5 | setosa |
| 1 | 4.9 | 3.0 | setosa |
| 2 | 4.7 | 3.2 | setosa |
| 3 | 4.6 | 3.1 | setosa |
| 4 | 5.0 | 3.6 | setosa |
| ... | ... | ... | ... |
| 50 | 7.0 | 3.2 | versicolor |
| 51 | 6.4 | 3.2 | versicolor |
| 52 | 6.9 | 3.1 | versicolor |
| 53 | 5.5 | 2.3 | versicolor |
| 54 | 6.5 | 2.8 | versicolor |
| 55 | 5.7 | 2.8 | versicolor |
| ... | ... | ... | |
| 145 | 6.7 | 3.0 | virginica |
| 146 | 6.3 | 2.5 | virginica |
| 147 | 6.5 | 3.0 | virginica |
| 148 | 6.2 | 3.4 | virginica |
| 149 | 5.9 | 3.0 | virginica |




For a new set of measurements:
Sepal length = **6.4**
Sepal width = **3.1**

Which species is it?

# MACHINE LEARNING: EXAMPLE

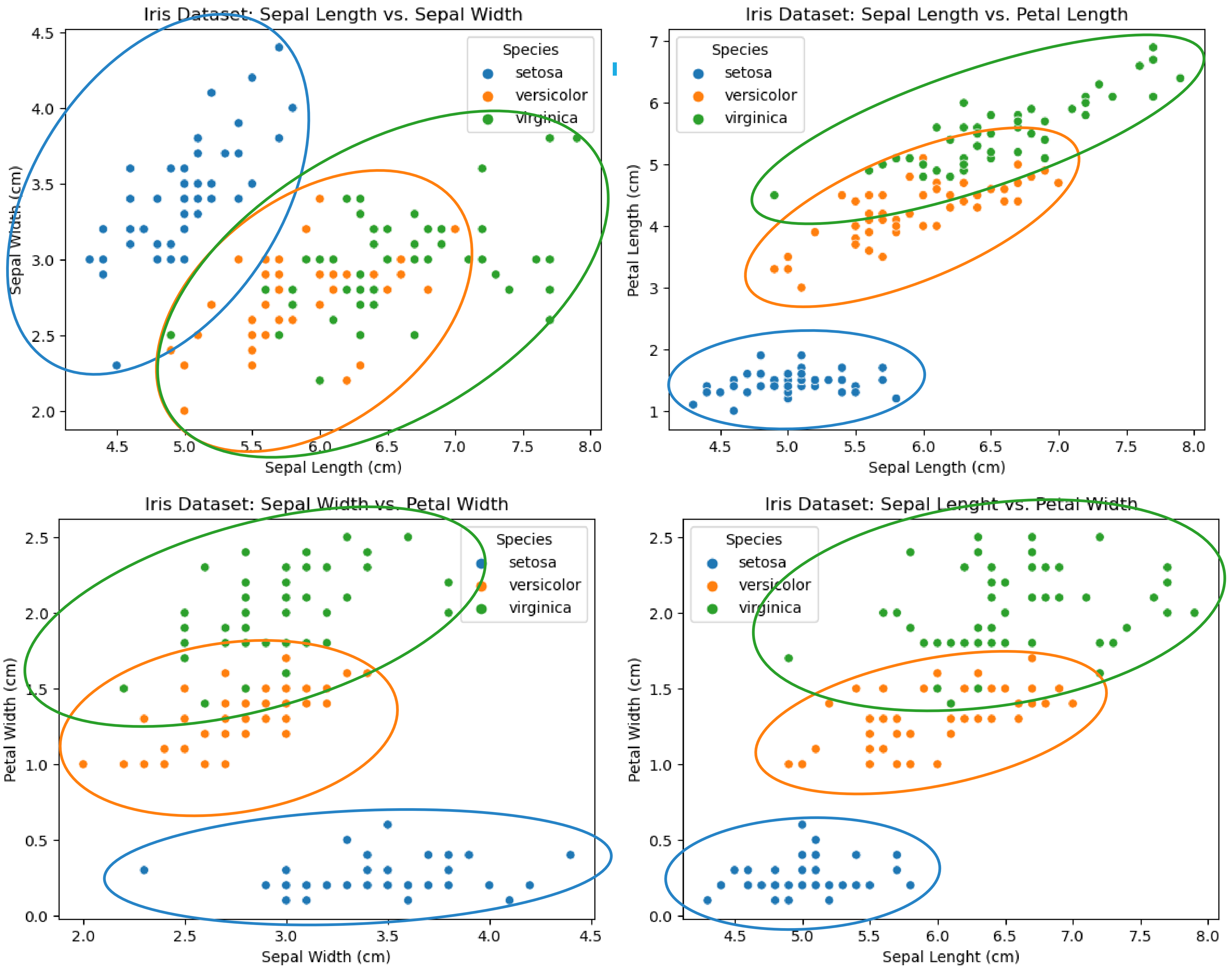
| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| … | … | … | … | … | … |
| 50 | 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| 51 | 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| 52 | 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| 53 | 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| 54 | 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| 55 | 5.7 | 2.8 | 4.5 | 1.3 | versicolor |
| … | … | … | … | … | |
| 145 | 6.7 | 3.0 | 5.2 | 2.3 | virginica |
| 146 | 6.3 | 2.5 | 5.0 | 1.9 | virginica |
| 147 | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| 148 | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| 149 | 5.9 | 3.0 | 5.1 | 1.8 | virginica |

Full dataset:

- 4 variables
- 3 species
- 150 samples

Given a new sample, how can we classify it?

Data exploration:



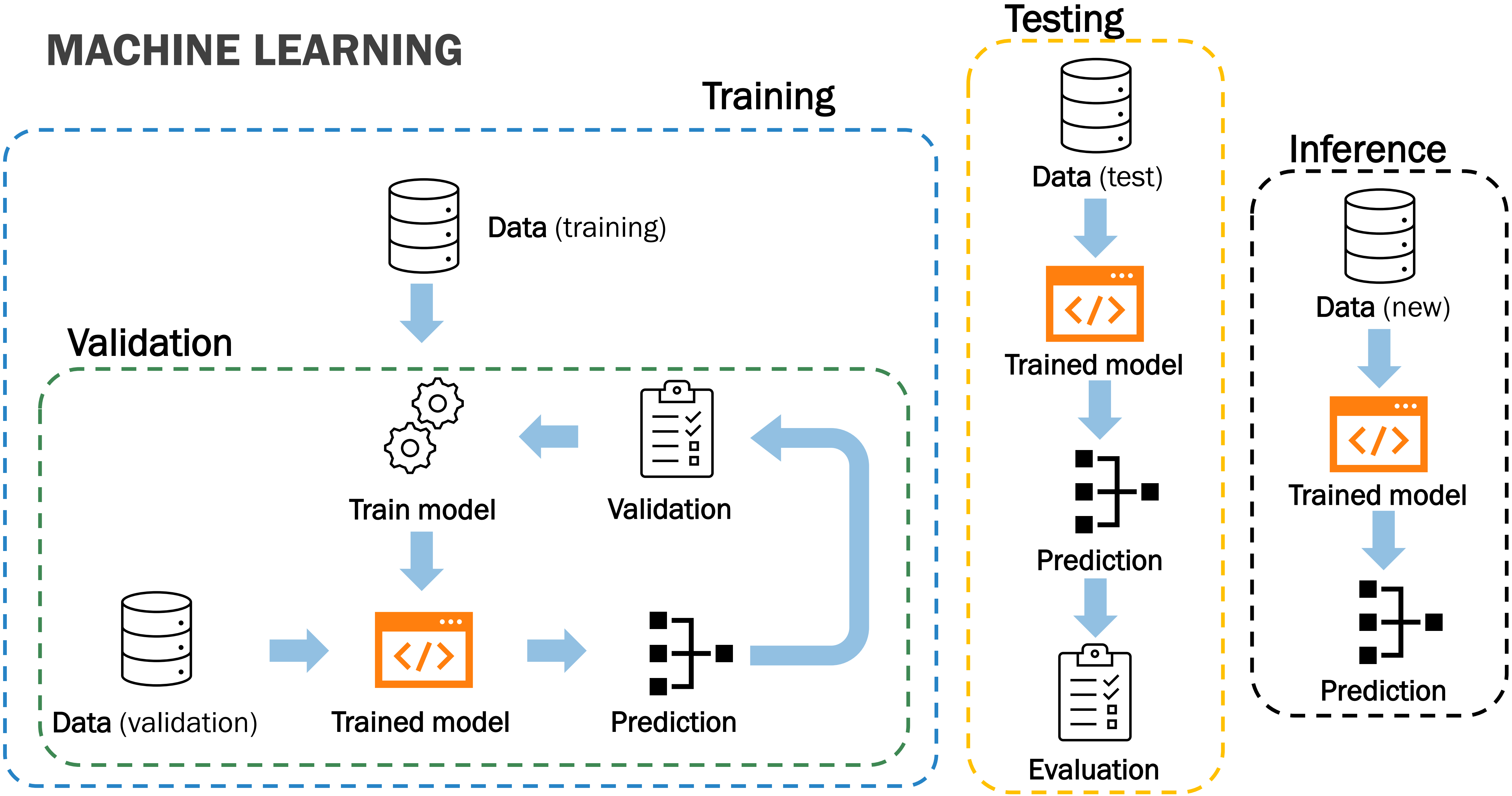

Given a new sample, how can we classify it?

**Machine learning**

# MACHINE LEARNING: DEFINITION

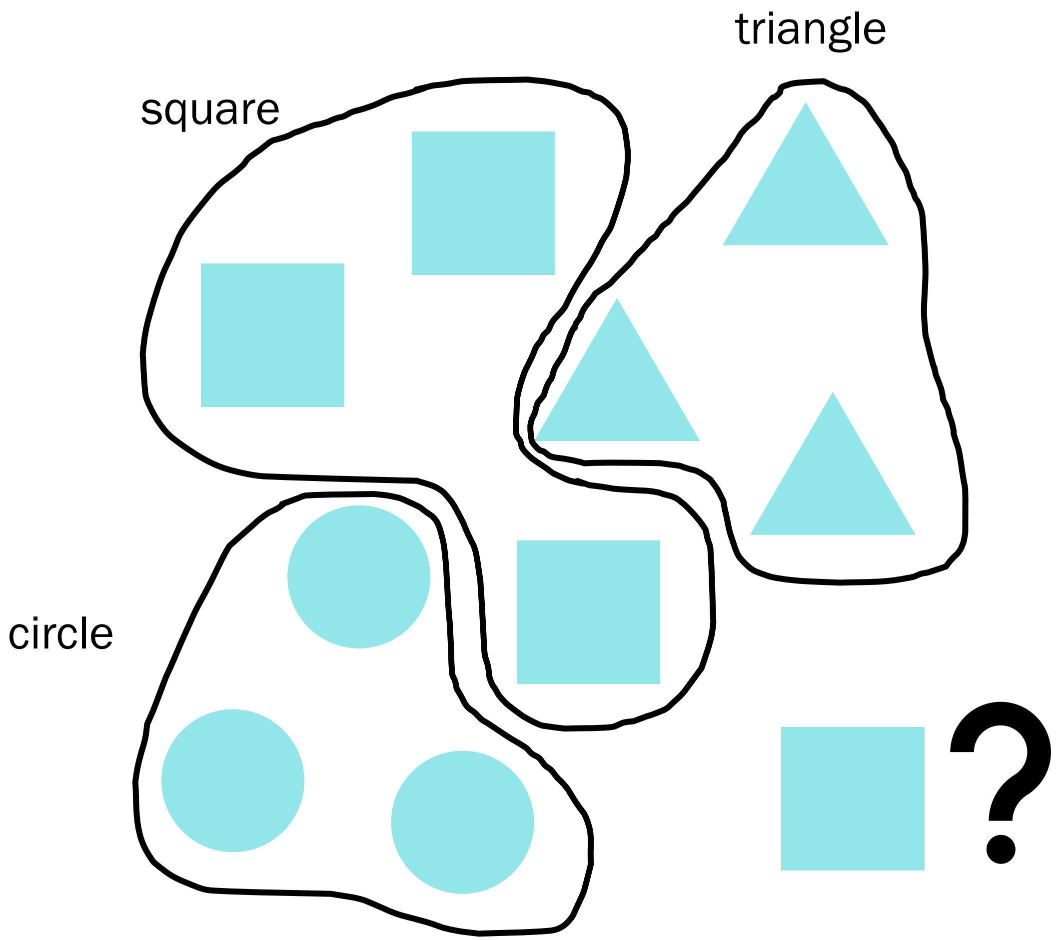
Subset of artificial intelligence (AI) **methods** that allow computers to **learn** from and make predictions or decisions based on **data**.

# MACHINE LEARNING

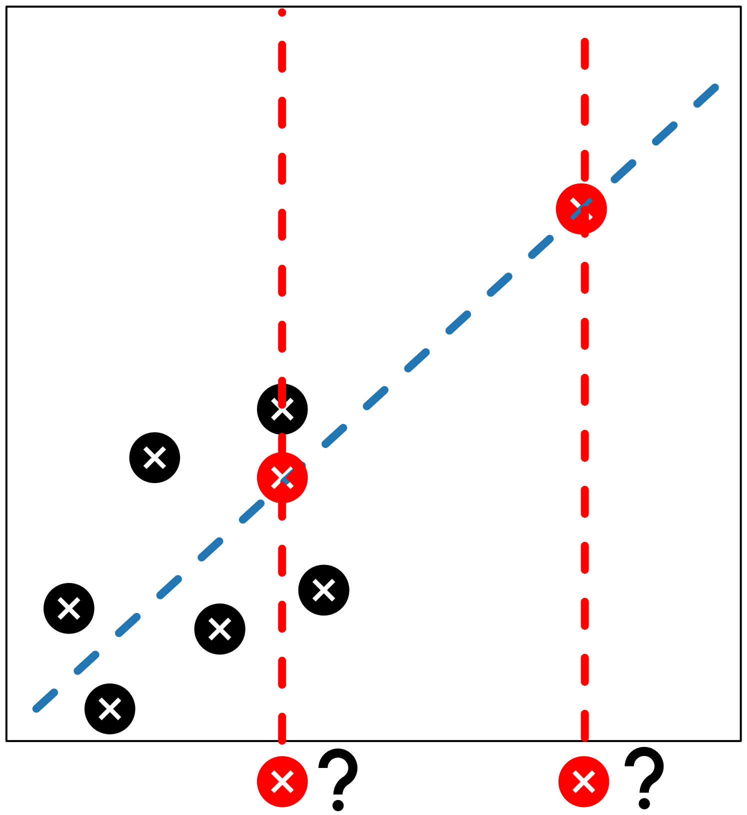


# CLASSIFICATION VS REGRESSION

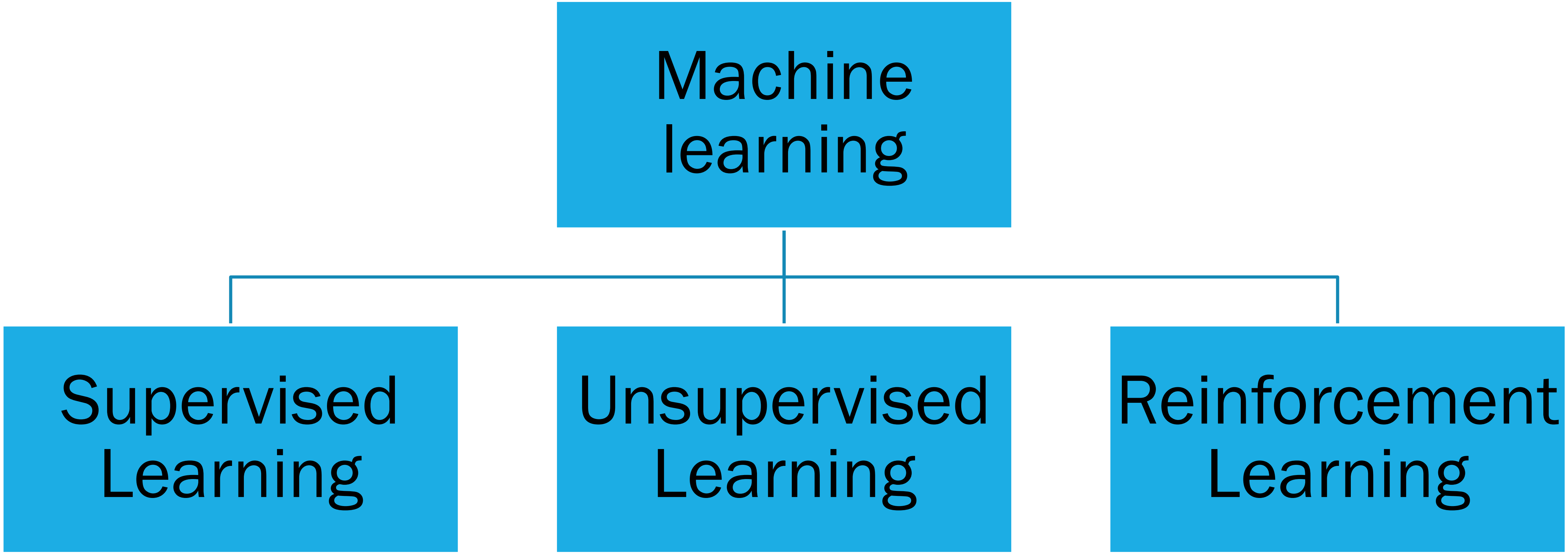
**Classification:** Assigns input data to a label based on patters or characteristics




**Regression:** models the relationship between a dependent variable and one or more independent variables to make predictions.



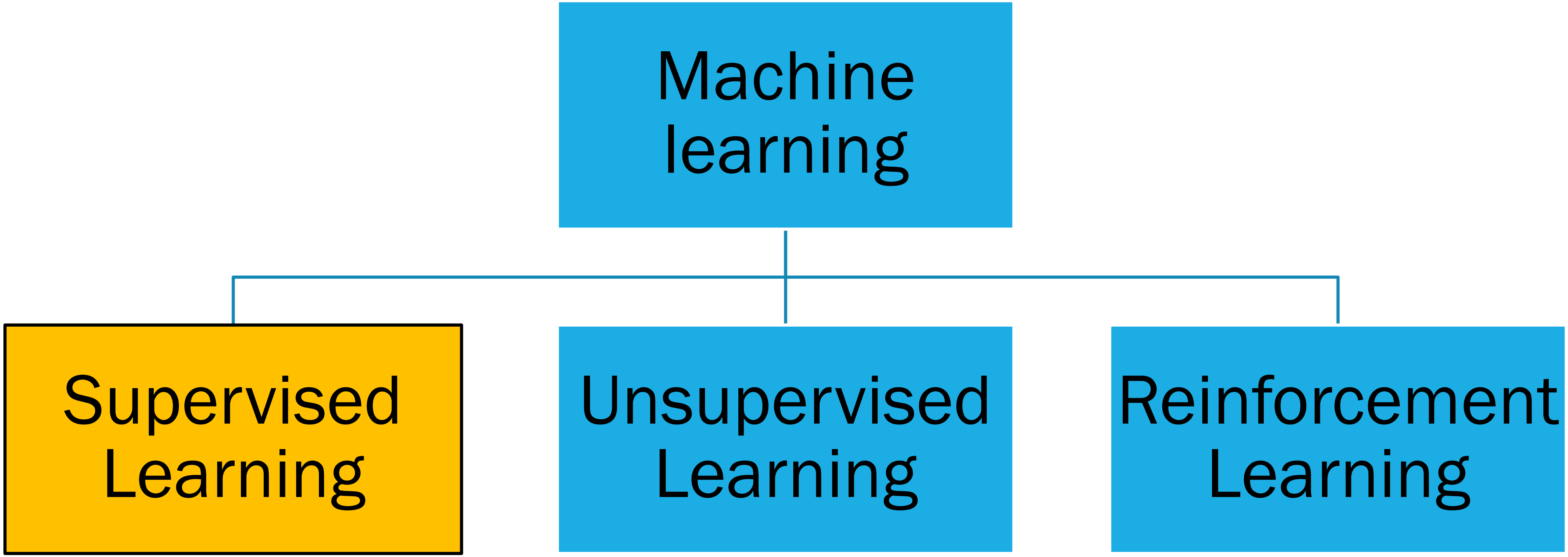
# TYPES OF MACHINE LEARNING




- Linear regression
- Decision trees

- K-means

# TYPES OF MACHINE LEARNING




- **Linear regression**
- Decision trees

- K-means

# SUPERVISED LEARNING

- The algorithm learns to make predictions by training on a **labelled** dataset.
- A label is the expected outcome of the prediction.
- Example using classification:

Input variables

Output

| sepal_length | sepal_width | species |
|---|---|---|
| 5.1 | 3.5 | setosa |
| 4.9 | 3.0 | setosa |
| 6.5 | 3.0 | virginica |
| 6.2 | 3.4 | virginica |
| 5.9 | 3.0 | virginica |

Labels / **Class**

In our Iris dataset, we know which species corresponds to each measurement. Setosa and Virginica are labels.

**When we train the algorithm, the algorithm learns the underlying patters between the input and the output.**

# SUPERVISED LEARNING: APPLICATIONS

## Regression

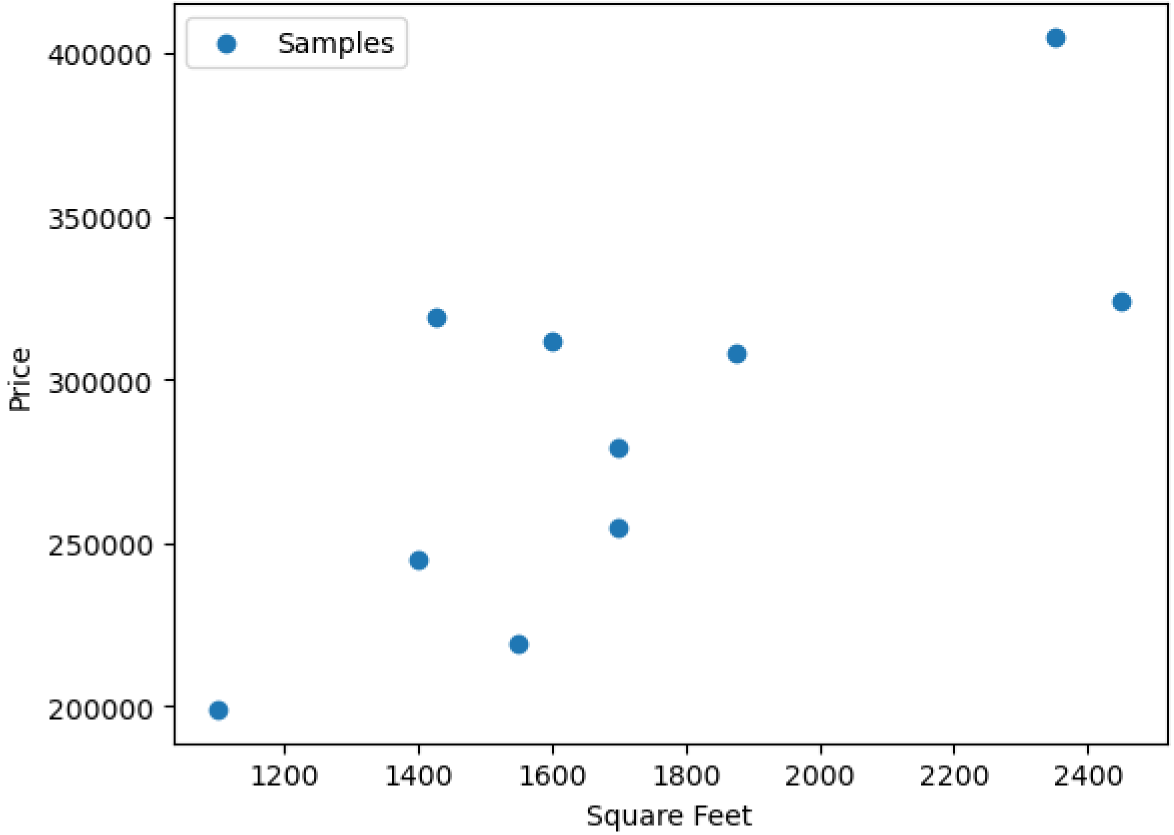
- Stock price prediction
- House price prediction
- Energy consumption prediction
- Weather prediction

## Classification

- Medical diagnosis
- Spam detection
- Credit scoring
- Handwriting recognition

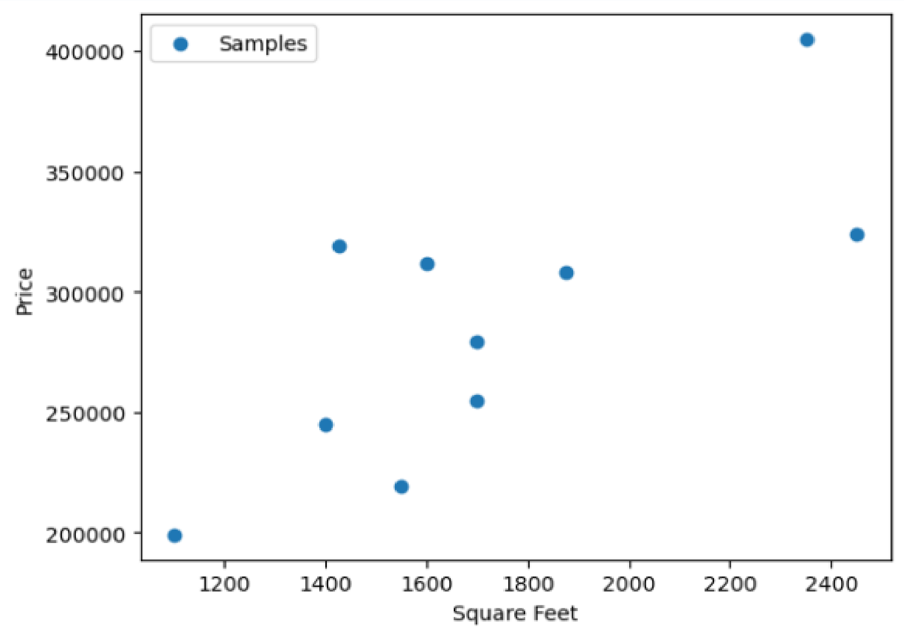
# REGRESSION: LINEAR REGRESSION - EXAMPLE

House prices: here we have some samples of previous house prices and the square feet:




Looking at the scatter plot, what is your **prediction** for a house with 2200 square feet?

# What is your price prediction for a house with 2200 square feet?




400K

0%

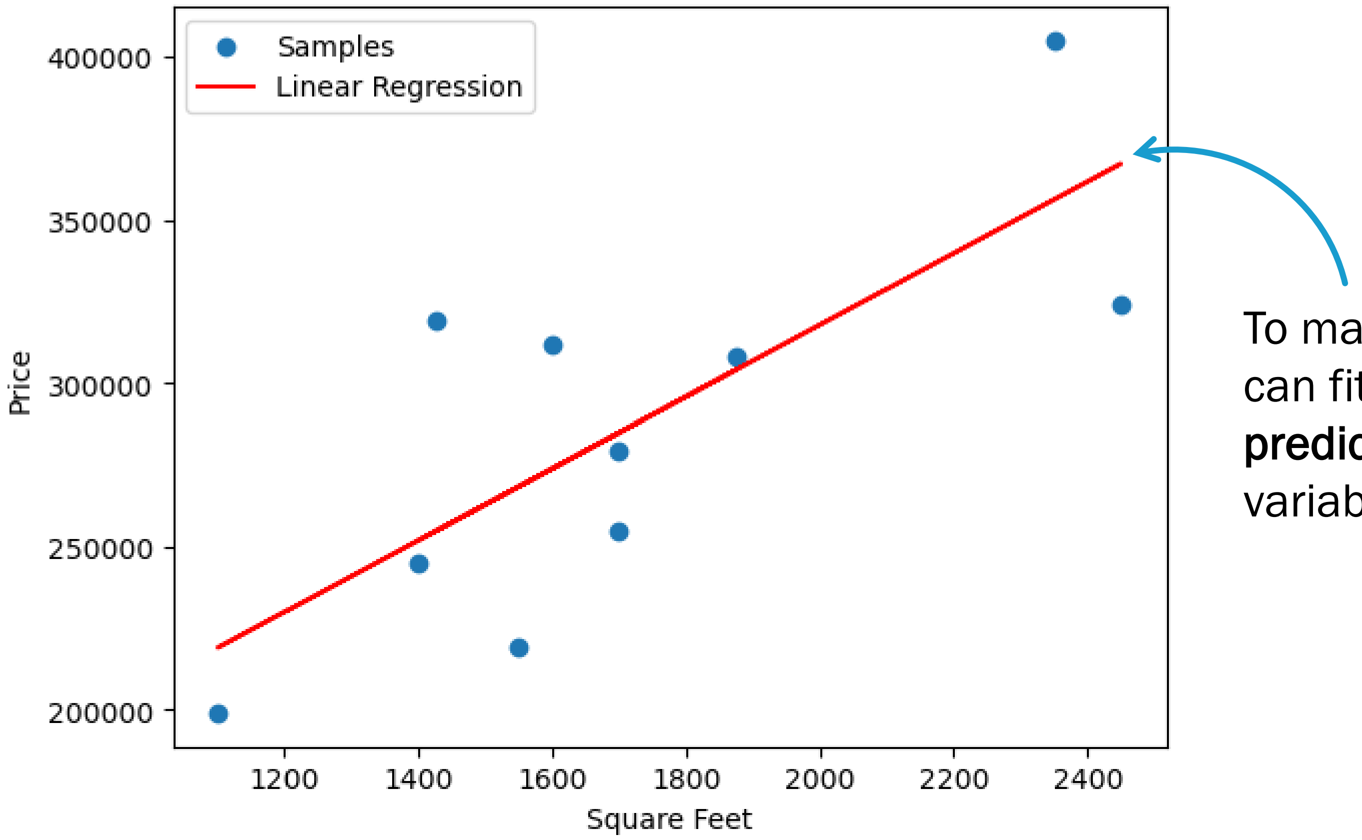
350K

0%

250K

0%

# REGRESSION: LINEAR REGRESSION




To make a simple prediction of house prices, we can fit a linear equation to the observed data to **predict** or **estimate** the value of a dependent variable.

# REGRESSION: LINEAR REGRESSION

- Finds the correlation between variables and enables to predict the continuous output variable based on one or more predictor variables.
- **Simple linear regression** (one independent variable): The model finds the linear relationship between the independent variable (x) and the dependent variable (y):

$$y = b_0 + b_1 x + \varepsilon \longrightarrow \text{Straight line}$$

Where:

$y$ is the dependent variable

$x$ is the independent variable

$b_0$ is the intercept (the point where the regression line intersects the y-axis

$b_1$ is the slope (how the dependent variable changes with a unit change in the independent variable)

$\varepsilon$ is the error term (residual) between the predicted and actual values

How do we find $b_0$ and $b_1$?

# LINEAR REGRESSION: FIT THE MODEL

How do we find $b_0$ and $b_1$?

- Typically, this is done by using the least squares method which tries to minimise the sum of the squared residuals (**cost function**):

$$min \sum(\underset{\substack{\uparrow \\ \text{True value} \\ (\textit{label})}}{y_1} - \underbrace{(b_0 + b_1 x_1)}_{\text{Predicted value}})^2$$

- Many forms of learning involve adjusting weights to minimise a loss using an optimisation technique such as **gradient descend.**

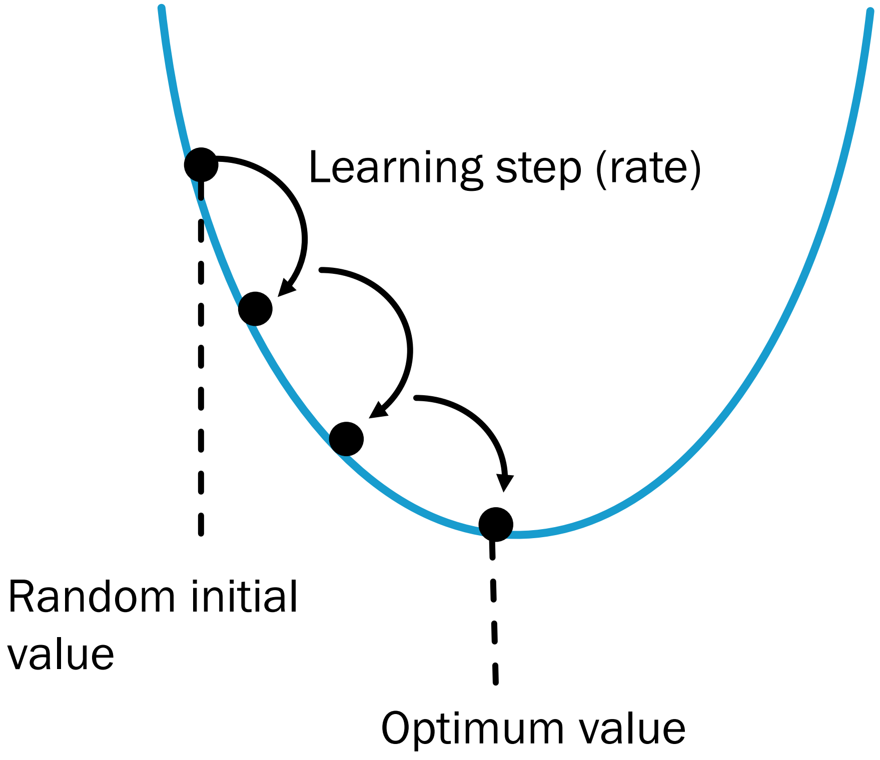
# GRADIENT DESCENT (OPTIMISATION)

- Iterative optimisation to find the minimum of a function (cost function/loss function) by adjusting the model parameters in the direction of the steepest descend.
- The idea is to update the parameters in the direction that minimises the cost function.
- Algorithm:

1. Initialise $b_0$ and $b_1$ with random values
2. While difference > threshold or there is no change (**convergence**) do:
   1. Compute how much the parameters need to change (by finding the partial derivatives of the cost function with respect to each parameter)
   2. Update parameters $b_0$ and $b_1$ using:

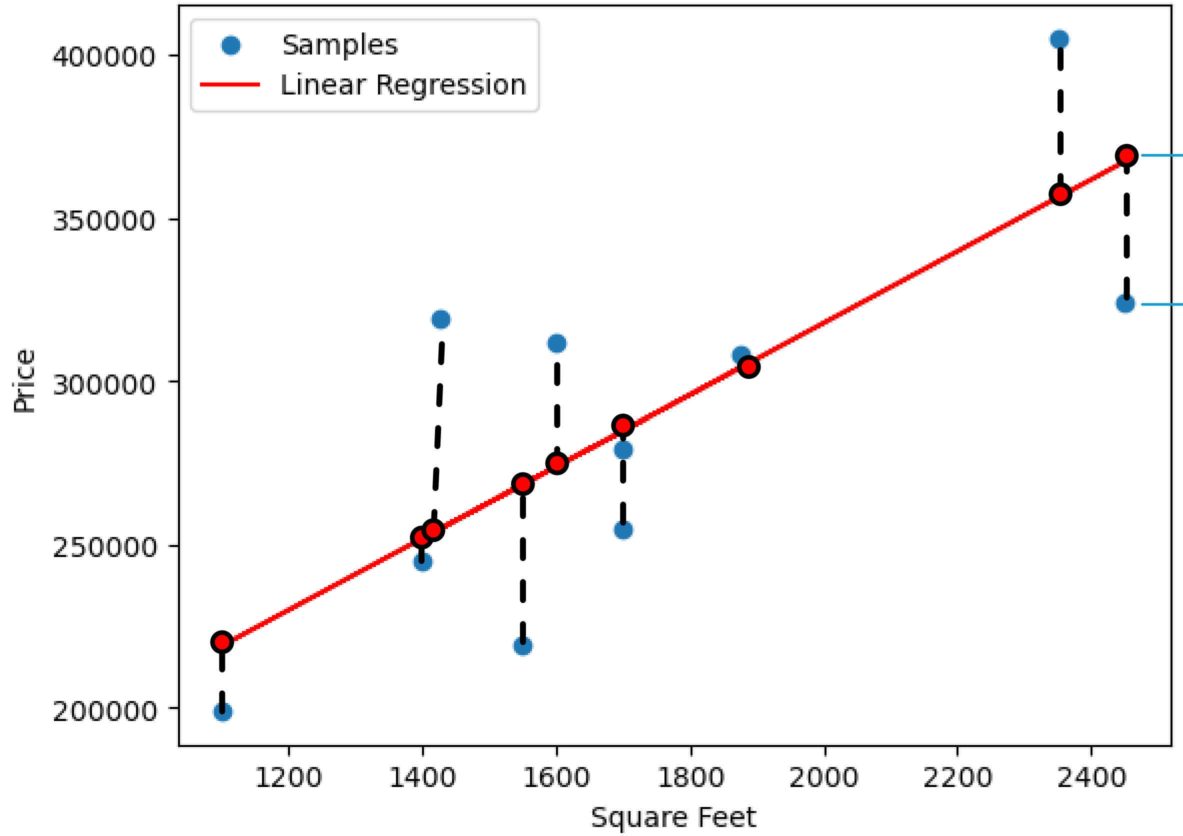
$$b_0 = b_0 - (LearningRate * \frac{\partial J}{\partial b_0})$$

$$b_1 = b_1 - (LearningRate * \frac{\partial J}{\partial b_1})$$



# LINEAR REGRESSION: EVALUATION

- How good is the model: we use a metric such as the Mean Squared Error (MSE)




Predicted value

True value

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

# LINEAR REGRESSION: CODE (PYTHON)

```
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Sample data - square footage and corresponding house prices
square_feet = np.array([1400, 1600, 1700, 1875, 1100, 1550, 2350, 2450, 1425, 1700])
house_prices = np.array([245000, 312000, 279000, 308000, 199000, 219000, 405000, 324000, 
319000, 255000])

# Reshape the data
square_feet = square_feet.reshape(-1, 1)

# Create and train the linear regression model
model = LinearRegression()
model.fit(square_feet, house_prices)

# Make predictions
predicted_prices = model.predict(square_feet)
```

# LINEAR REGRESSION: RESEARCH

**COVID-19 PANDEMIC CHALLENGES, COPING STRATEGIES AND RESILIENCE AMONG HEALTHCARE WORKERS: A MULTIPLE LINEAR REGRESSION ANALYSIS**

**Anita Mfuh Y. Lukong[1] and Yahaya Jafaru[2]**

[1]Senior Lecturer, PhD, RN Department of Nursing Science, College of Health Sciences, Federal University Birnin Kebbi, P.M.B. 1157, Birnin Kebbi, Kebbi State, Nigeria.

E-mail Address: lukonganita@yahoo.com

[2]Lecturer, MSc, RN Department of Nursing Science, College of Health Sciences, Federa University Birnin Kebbi, P.M.B. 1157, Birnin Kebbi, Kebbi State, Nigeria.

ELSEVIER

Energy

Volume 225, 15 June 2021, 120270

## Modelling industry energy demand using multiple linear regression analysis based on consumed quantity of goods

Mohamed Maaouane [a], Smail Zouggar [a], Goran Krajačić [b], Hassan Zahboune [a]

# REGRESSION

If you are interested in other algorithms, you can explore the following ones:

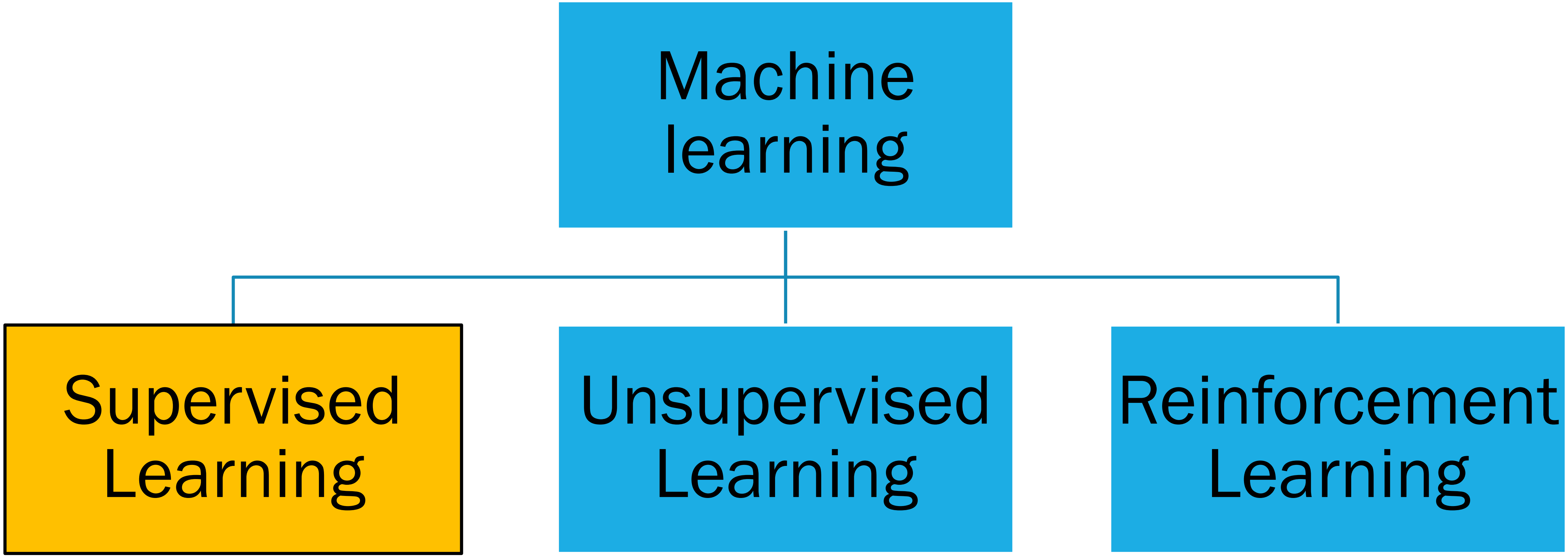
- Multiple linear regression
- Logistic regression

# What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

# TYPES OF MACHINE LEARNING

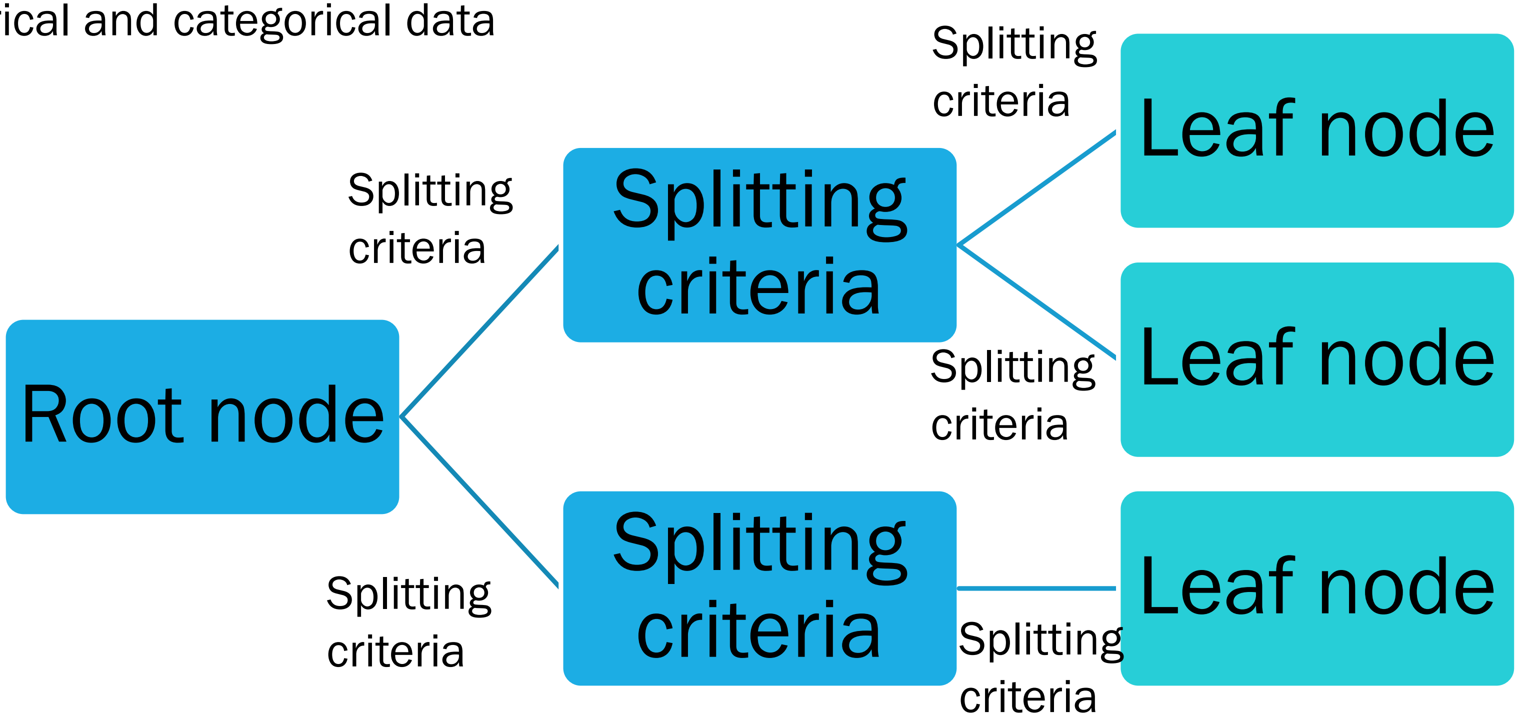



- Linear regression
- **Decision trees**

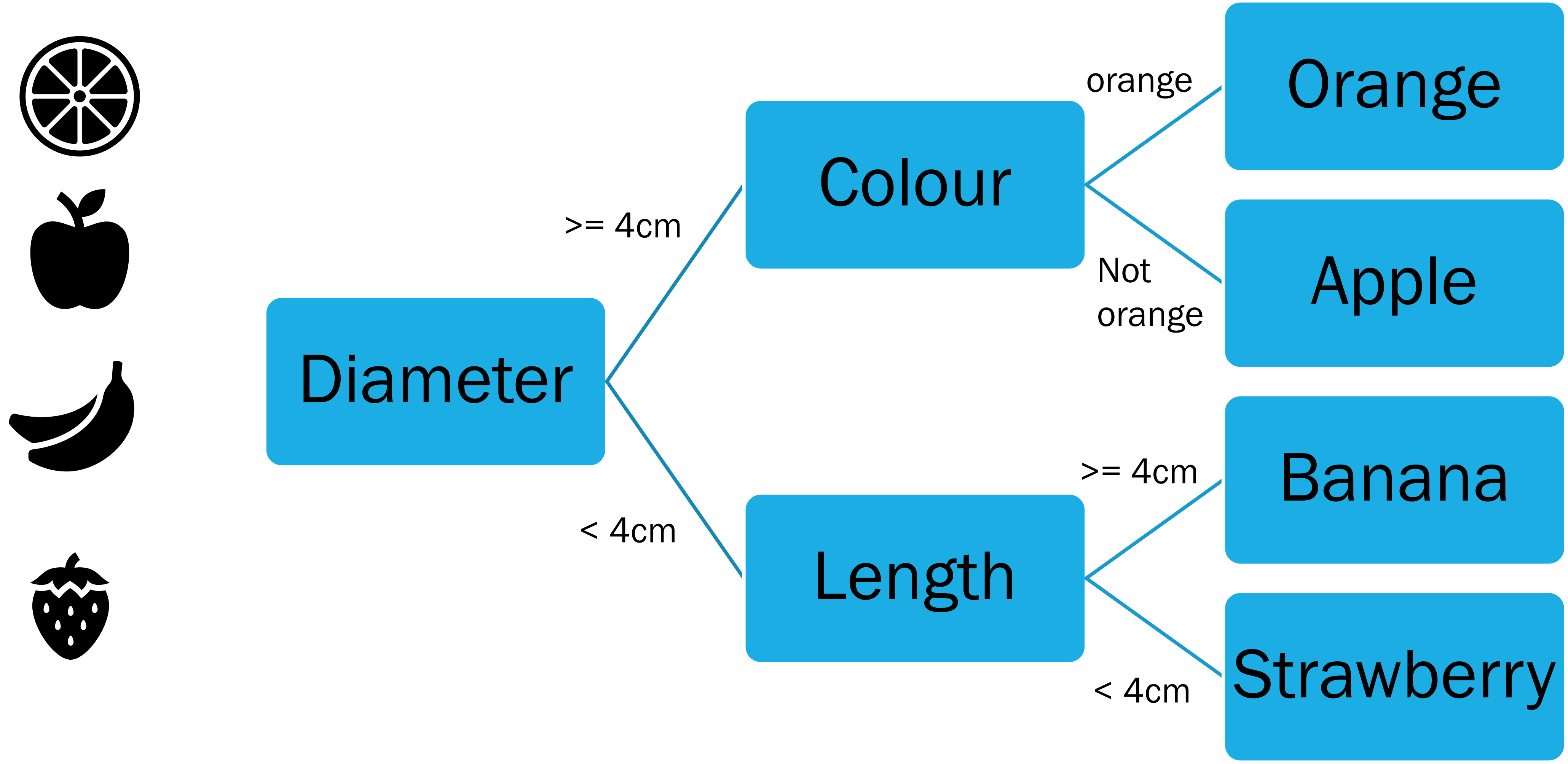
- K-means

# CLASSIFICATION: DECISION TREES

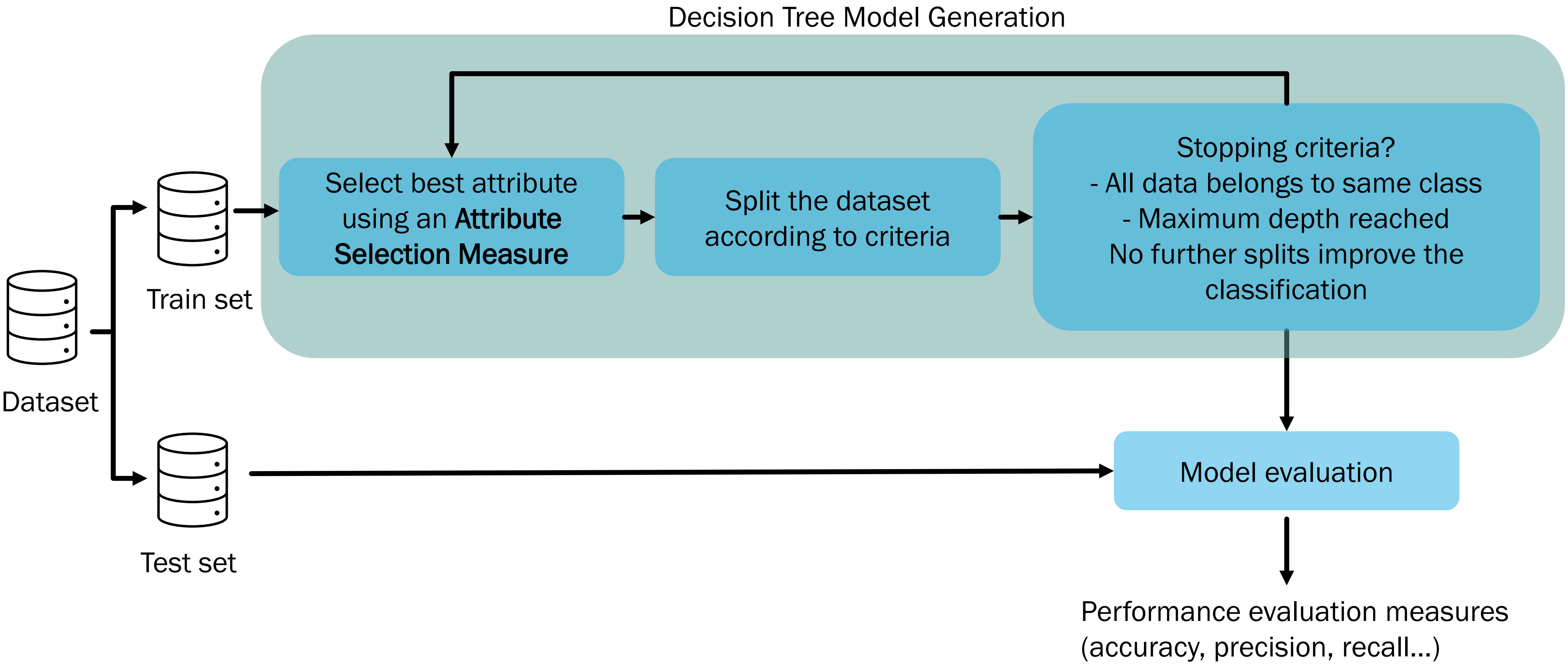
- A decision tree is a supervised machine learning algorithm used for classification or regression tasks.
- Makes decisions based on the input data to provide a prediction.
- Makes decisions by asking a series of questions based on the features of the input data to output a prediction.
- They can handle both numerical and categorical data



# DECISION TREES: EXAMPLE



# DECISION TREES: HOW IT WORKS?



# DECISION TREES: ATTRIBUTE SELECTION MEASURE

- **Gini index:** It quantifies the likelihood of misclassifying a randomly chosen sample if it were randomly classified.
- In other words, it measures the disorder in a set of data.
- Min value = 0; max value = 0.5;

$$Gini(D) = 1 - (p_1^2 + p_2^2)$$

Dataset

Proportion of samples in class 1

Proportion of samples in class 2

Gini index is used to evaluate the impurity of a node before and after the data split. The **goal is to minimise the Gini index** by selecting the feature and split point that result in the purest child nodes.

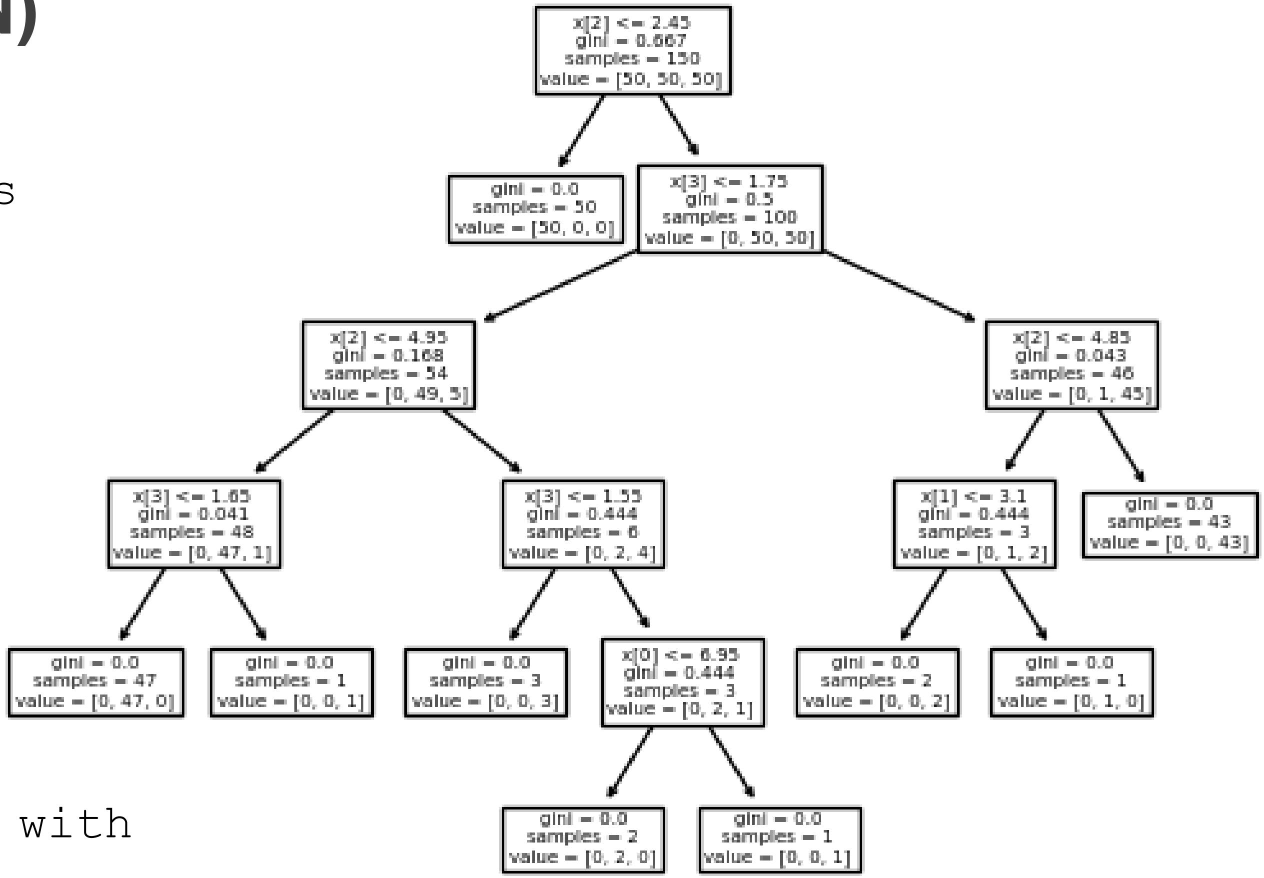
# DECISION TREES: CODE (PYTHON)

```
from sklearn.datasets import load_iris
from sklearn import tree

# Load Iris dataset
iris = load_iris()

# Train model
X, y = iris.data, iris.target
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, y)

# Once trained, you can plot the tree with
the plot_tree function:
tree.plot_tree(clf)
```



# RESEARCH

Journals & Magazines > IEEE Access > Volume: 9

## Experimental Setup for Online Fault Diagnosis of Induction Machines via Promising IoT and Machine Learning: Towards Industry 4.0 Empowerment

**Publisher: IEEE** Cite This PDF

Minh-Quang Tran ; Mahmoud Elsisi ; Karar Mahmoud ; Men… All A

In this study, three effective ensemble methods machine learning techniques including **decision tree (DT), random forest (RF), and extreme gradient boosting (XGBoost)** algorithms are used to distinguish different bearing conditions.

## Reliable Industry 4.0 Based on Machine Learning and IoT for Analyzing, Monitoring, and Securing Smart Meters

by Mahmoud Elsisi [1,2], Karar Mahmoud [3,4], Matti Lehtonen [3] and Mohamed M. F. Darwish [2,3,*]

1 Industry 4.0 Implementation Center, Center for Cyber–Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

2 Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt

3 Department of Electrical Engineering and Automation, Aalto University, FI-00076 Espoo, Finland

4 Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt

* Author to whom correspondence should be addressed.

*Sensors* **2021**, *21*(2), 487; https://doi.org/10.3390/s21020487

**Received: 22 December 2020 / Revised: 1 January 2021 / Accepted: 10 January 2021 / Published: 12 January 2021**

(This article belongs to the Section Intelligent Sensors)

Download Browse Figures Versions Notes

**Abstract**

The modern control infrastructure that manages and monitors the communication between the smart machines represents the most effective way to increase the efficiency of the industrial environment, such as smart grids. The cyber-physical systems utilize the embedded software and internet to connect and control the smart machines that are addressed by the internet of things (IoT). These cyber-physical systems are the basis of the fourth industrial revolution which is indexed by industry 4.0. In particular, industry 4.0 relies heavily on the IoT and smart sensors such as smart energy meters. The reliability and security represent the main challenges that face the industry 4.0 implementation. This paper introduces a new infrastructure based on machine learning to analyze and monitor the output data of the smart meters to investigate if this data is real data or fake. The fake data are due to the hacking and the inefficient meters. The industrial environment affects the efficiency of the meters by temperature, humidity, and noise signals. Furthermore, the proposed infrastructure validates the amount of data loss via communication channels and the internet connection. The decision tree is utilized as an effective machine learning algorithm to carry out both regression and classification for the meters' data. The data monitoring is carried out based on the industrial digital twins' platform. The proposed infrastructure results provide a reliable and effective industrial decision that enhances the investments in industry 4.0.

**Keywords:** smart systems; industry 4.0; internet of things; machine learning

# CLASSIFICATION

If you are interested in other algorithms, you can explore the following ones:

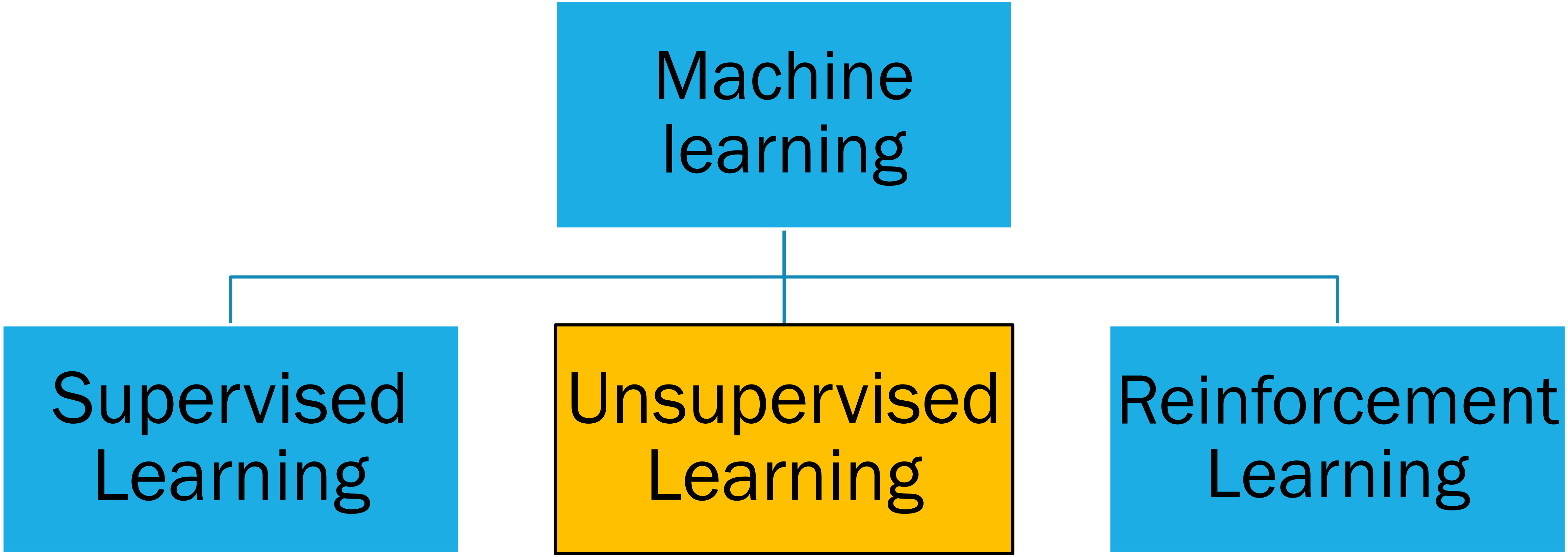
- Bayesian Networks
- Random forests
- K-Nearest Neighbours
- Support Vector Machines

# What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

# TYPES OF MACHINE LEARNING




- Linear regression
- Decision trees

- K-means

# UNSUPERVISED LEARNING

- The algorithm find patters in a dataset without labels.
- Usually used for:
  - Clustering: group similar data together
  - Dimensionality reduction: reduce the number of features in a dataset
- Example using clustering:

Input variables

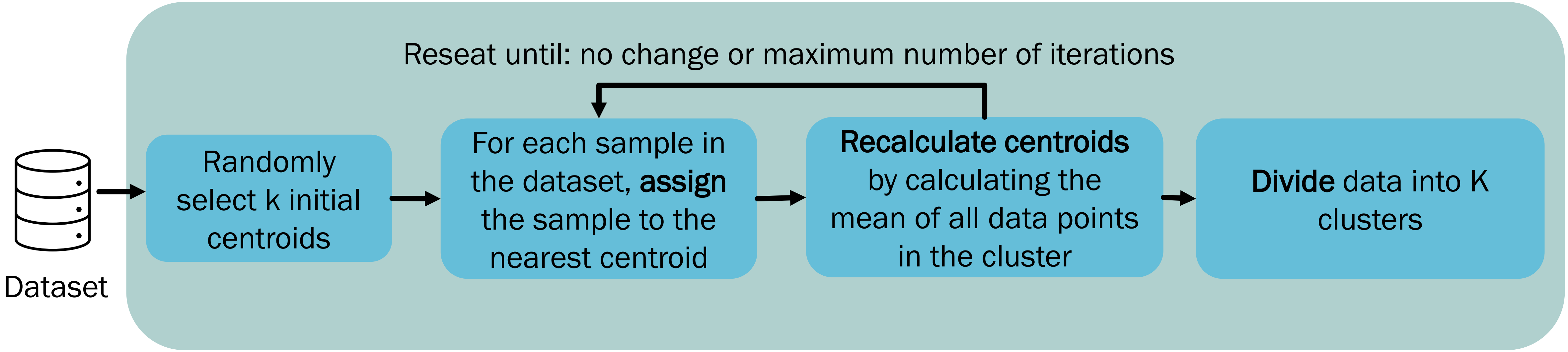
Output

| sepal_length | sepal_width | species |
|---|---|---|
| 5.1 | 3.5 | ~~setosa~~ |
| 4.9 | 3.0 | ~~setosa~~ |
| 6.5 | 3.0 | ~~virginica~~ |
| 6.2 | 3.4 | ~~virginica~~ |
| 5.9 | 3.0 | ~~virginica~~ |

# UNSUPERVISED LEARNING: K-MEANS

- Popular algorithm used to group data (samples) into clusters.
- **K:** stands for the number of cluster (groups) you want to identify (e.g., K=3 → 3 groups)
- **Cluster centre:** each cluster has a centroid (centre) which the algorithm finds such that better represent the cluster.

**K-means algorithm**



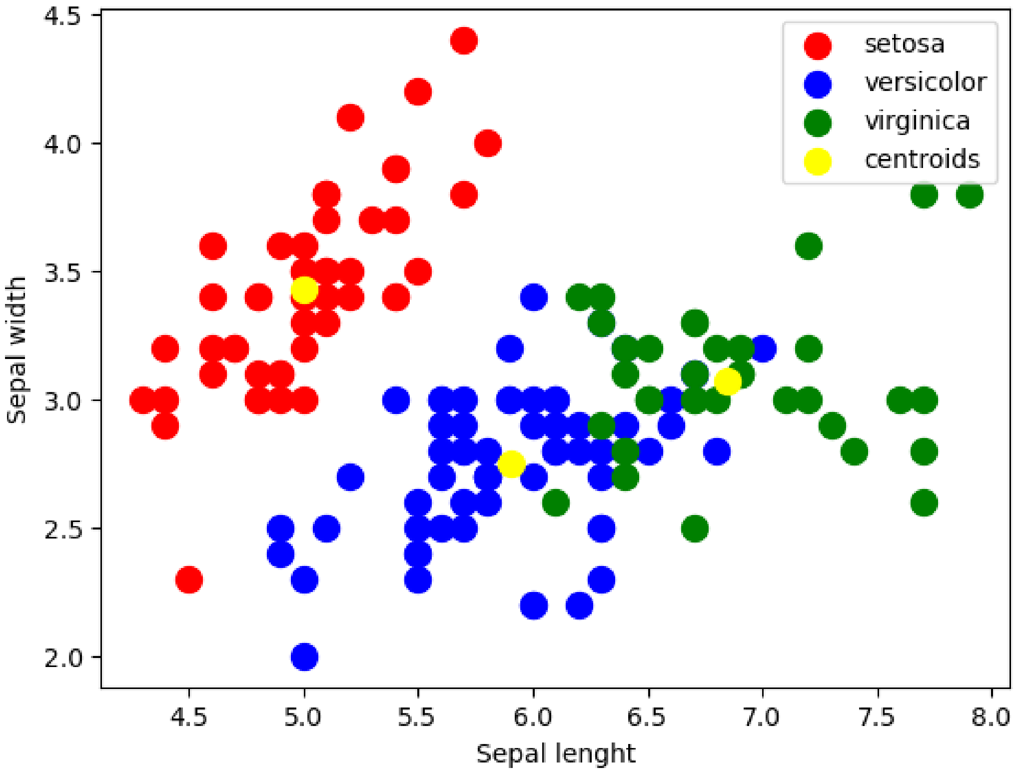
# K-MEANS: CODE (PYTHON)

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import load_iris

# Load Iris dataset
iris = load_iris()
x = iris.data

# Create and fit the model
model = KMeans(n_clusters = 3, max_iter = 300, n_init = 10)
y = model.fit_predict(x)

# Visualise the clusters (groups)
plt.scatter(x[y == 0, 0], x[y == 0, 1], s = 100, c = 'red', label = 'setosa')
plt.scatter(x[y == 1, 0], x[y == 1, 1], s = 100, c = 'blue', label = 'versicolor')
plt.scatter(x[y == 2, 0], x[y == 2, 1], s = 100, c = 'green', label = 'virginica')
plt.scatter(model.cluster_centers_[:,0], model.cluster_centers_[:,1], s = 100, c =
'yellow', label = 'centroids')
plt.legend()
```



# UNSUPERVISED LEARNING: APPLICATIONS

- **Exploratory data analysis:** Uncover hidden patterns (e.g., detect anomalies in credit card transactions).
- **Anomaly detection:** Detect anomalies or outliers in datasets (e.g., cyberattacks).
- **Reducing data complexity**: Simplify data to make it more interpretable (e.g., simplify the high-dimensional gene expression data).
- **Feature engineering**: To generate new features of group similar features (e.g. identify genes that are linked to a disease).

# RESEARCH

Open Access Article

## Identification of Student Behavioral Patterns in Higher Education Using K-Means Clustering and Support Vector Machine

by Nur Izzati Mohd Talib, Nazatul Aini Abd Majid * and Shahnorbanun Sahran

Center for Artificial Intelligence Technology, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia

* Author to whom correspondence should be addressed.

*Appl. Sci.* **2023**, *13*(5), 3267; https://doi.org/10.3390/app13053267

ELSEVIER

Smart Agricultural Technology

Volume 3, February 2023, 100081

## Diagnosis of grape leaf diseases using automatic *K*-means clustering and machine learning

Seyed Mohamad Javidan [a], Ahmad Banakar [a], Keyvan Asefpour Vakilian [b], Yiannis Ampatzidis [c]

Conferences > 2023 13th International Confe...

## Customer Segmentation of E-commerce data using K-means Clustering Algorithm

Publisher: IEEE Cite This PDF

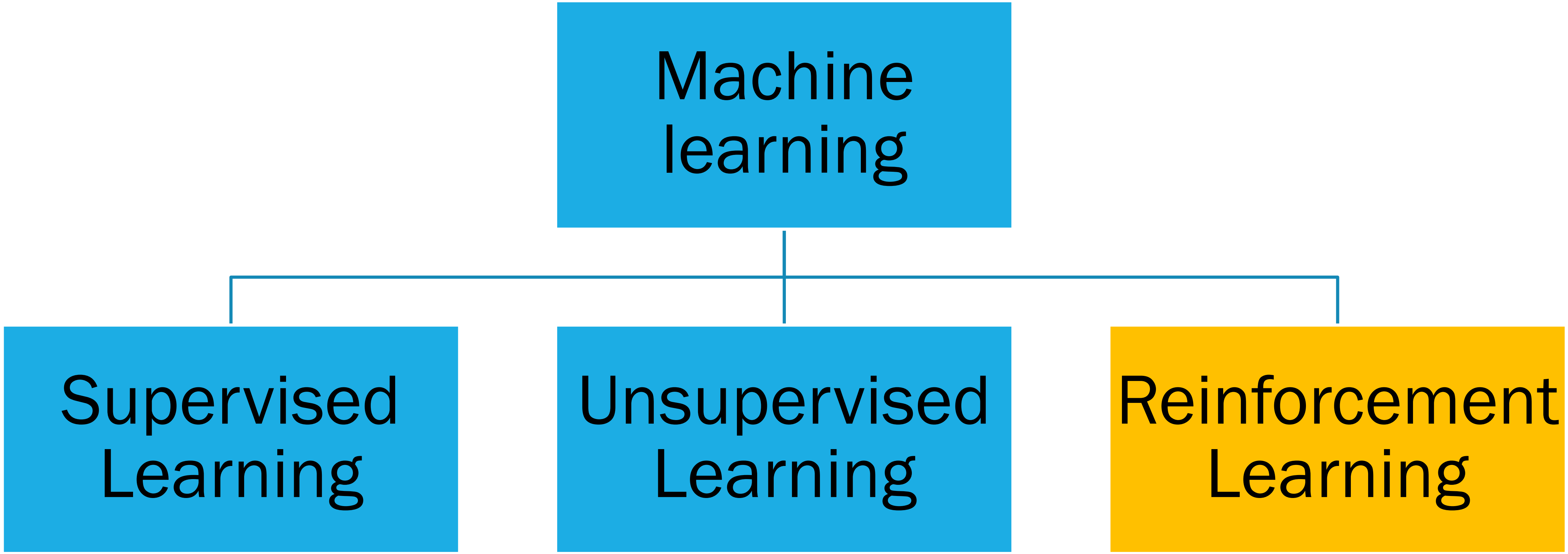
Lucky Rajput ; Shailendra Narayan Singh All Authors

# What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

## TYPES OF MACHINE LEARNING



# REINFORCEMENT LEARNING

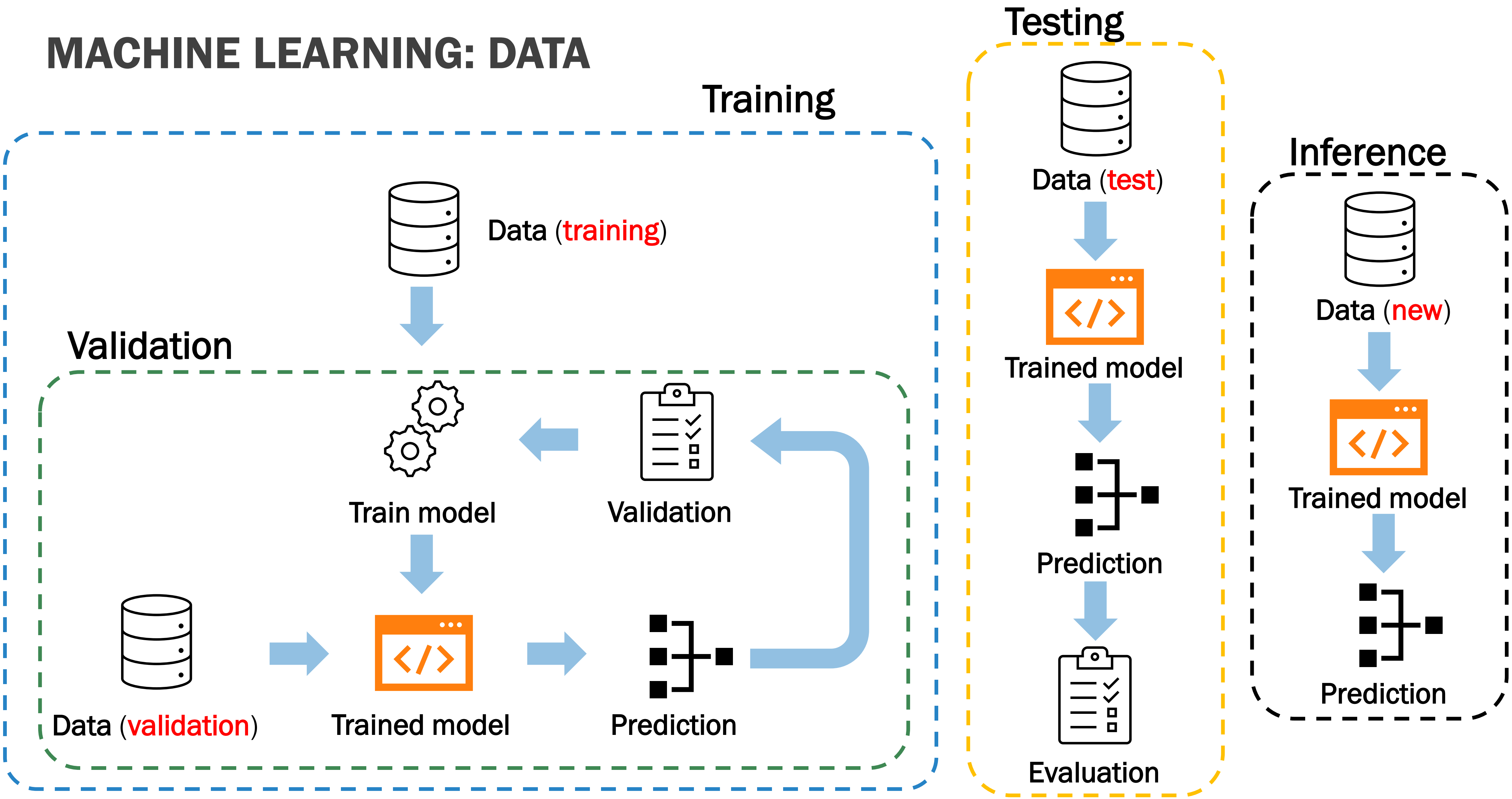
- Reinforcement learning will be covered in detail during **week 11.**
- A type or machine learning algorithm where an agent interacts with an environment and learns to take actions to maximise a reward.
- Machine learning can also be used in reinforcement learning.

# DATA

# DATA

- Data refers to the information that we use to build, train and evaluate machine learning models.
- Data consists of features or variables.
- There are two types of variables
  - **Numerical**: variable with continuous values (e.g., 1.1, 1.2, 1.11, etc)
  - **Categorical**: variables with categorical values (e.g., ‘blue’, ‘red’, etc)
- Datasets are often divided into sets:
  - Train set: data used to train the models.
  - Validation set: data used to validate and fine-tune the model.
  - Testing set: data used to evaluate the model’s performance.

# MACHINE LEARNING: DATA



# DATA: SPLIT

How do we split the data?

| Training set | Validation set | Test set | |
|---|---|---|---|
| 80% | 10% | 10% | (small datasets) |
| 60% | 20% | 20% | (larger datasets) |

- Know your data:

**Balanced dataset**

50% data
Setosa

50% data
Virginica

**Imbalanced dataset**

80% data
Setosa

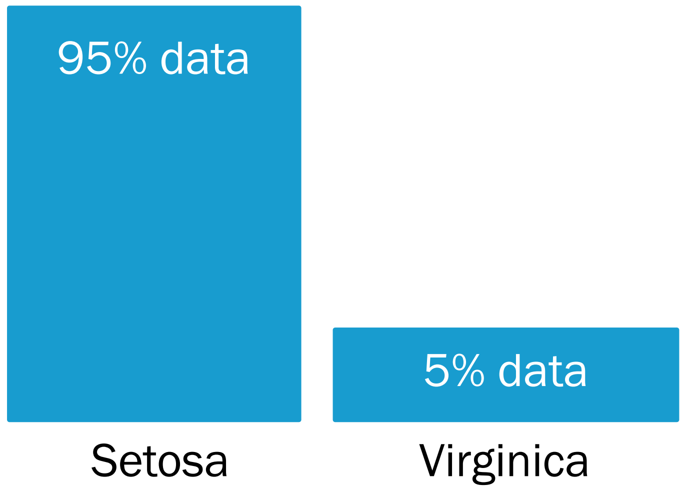
20% data
Virginica

# Do you think an accuracy of 95% means that the model is accurate?

| | |
|---|---|
| Yes | 0% |
| No | 0% |
| Depends | 0% |

# PROBLEMS WITH IMBALANCED DATASETS: ACCURACY

Imbalanced dataset (100 samples)




What is for you a good accuracy?

If we predict everything as a Setosa....

$$accuracy = \frac{number\ correct\ samples}{total\ number\ of\ samples}$$

$$accuracy = \frac{95}{100} = \mathbf{0.95\ (95\%!!)}$$

This model would have a 95% accuracy even if predicts everything as a cat.

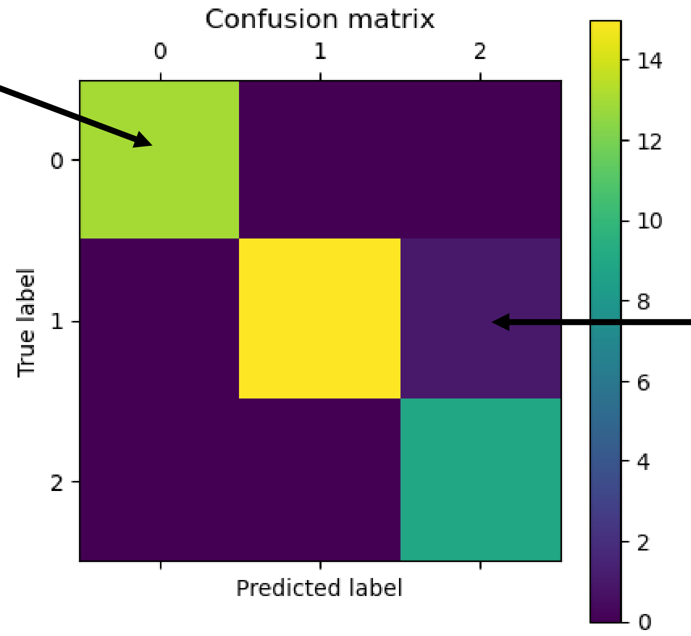
Remember:
1) Know your data
2) Use and report accuracy for each class

# CONFUSION MATRIX

- A confusion matrix is a method to summarise the performance of a classification algorithm using a table (matrix).
- True positive (**TP**): The number of samples correctly predicted as the positive class.
- False positive (**FP**): The number of samples incorrectly predicted as the positive class.
- True negative (**TN**): The number of samples correctly predicted as the negative class.
- False negative (**FN**): The number of samples incorrectly predicted as the negative class.

| | Predicted positive | Predicted negative |
|---|---|---|
| Actual positive | **TP** | **FN** |
| Actual negative | **FP** | **TN** |

Samples with label ‘0’ correctly predicted as ‘0’




Samples with label ‘1’ incorrectly predicted as ‘2’

# METRICS

Several other evaluation metrics are also used:

- **Accuracy:** Ratio of correctly predicted samples (TP+TN) to the total number of samples.
- **Precision:** Ratio of TP to total number of samples.
- **Recall:** Ratio of TP to the total number of samples that are positives.
- **Specificity:** Ratio of true negatives to the total number of samples that are negative.
- **F1 score:** Mean of precision and recall.

# What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

# TUTORIAL

**6COSC020W Applied AI**
Tutorial Week 7: Machine Learning

Aim:

- Review machine learning concepts.
- Experiment with TensorFlow and machine learning.
- Train and evaluate machine learning models for classification and regression.
- Explore new problems and datasets.

**1.- Definitions and concepts:**

**Question 1)** What are the three different types of machine learning? List and describe them using your own words.

**Question 2)** You have been asked to implement a system that tells us the expected the final grade of our students based on their results from year 4 and year 5. Which algorithm can you use?

**Question 3)** You have been asked to implement a system that given information about customer purchases you want to classify them in unknown buying behaviours or preferences.

**Question 4)** Explain using your own words the K-means algorithm.

**2.- Tutorial_Week7.ipynb.**

1) Download the file *Tutorial_Week7.ipynb* from Blackboard.
2) Open Anaconda and launch Jupyter Notebook.
3) Select and open Tutorial_Week7.ipynb and work through the Jupyter Notebook. Answer the questions and do the tasks.

## Applied AI - Machine Learning tutorial

Work through the notebook and answer the questions

## Regression (prediction) : Linear Regression

### 1.- Load and explore the dataset

In [ ]:
```
import seaborn as sns
tips = sns.load_dataset("tips")
```

Have a look at the data in the Tips dataset here: https://github.com/mwaskom/seaborn-data/blob/master/tips.csv

**Question: How many variables does the dataset have? Which variales are numerical and which variables are categorical?**

**Question: How many samples do you have in the dataset?**

Hint: you can use the .info or .count functions here.

### 2.- Visualise data from dataset

We will now plot two of the variables: total_bill and tip:

In [ ]:
```
import matplotlib.pyplot as plt
sns.scatterplot(x="total_bill", y="tip", data=tips, color='blue', label='Data')
plt.xlabel('Total Bill')
plt.ylabel('Tip')
plt.legend()
plt.title('Tips dataset')
plt.show()
```

## After session: What do you think is machine learning?

Nobody has responded yet.

Hang tight! Responses are coming in.

# What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

# ACKNOWLEDGEMENTS

Acknowledgements: This lecture was inspired by the following material:

- Machine Learning (Applied AI 22/23 – Dr. Hamed Hamzeh)