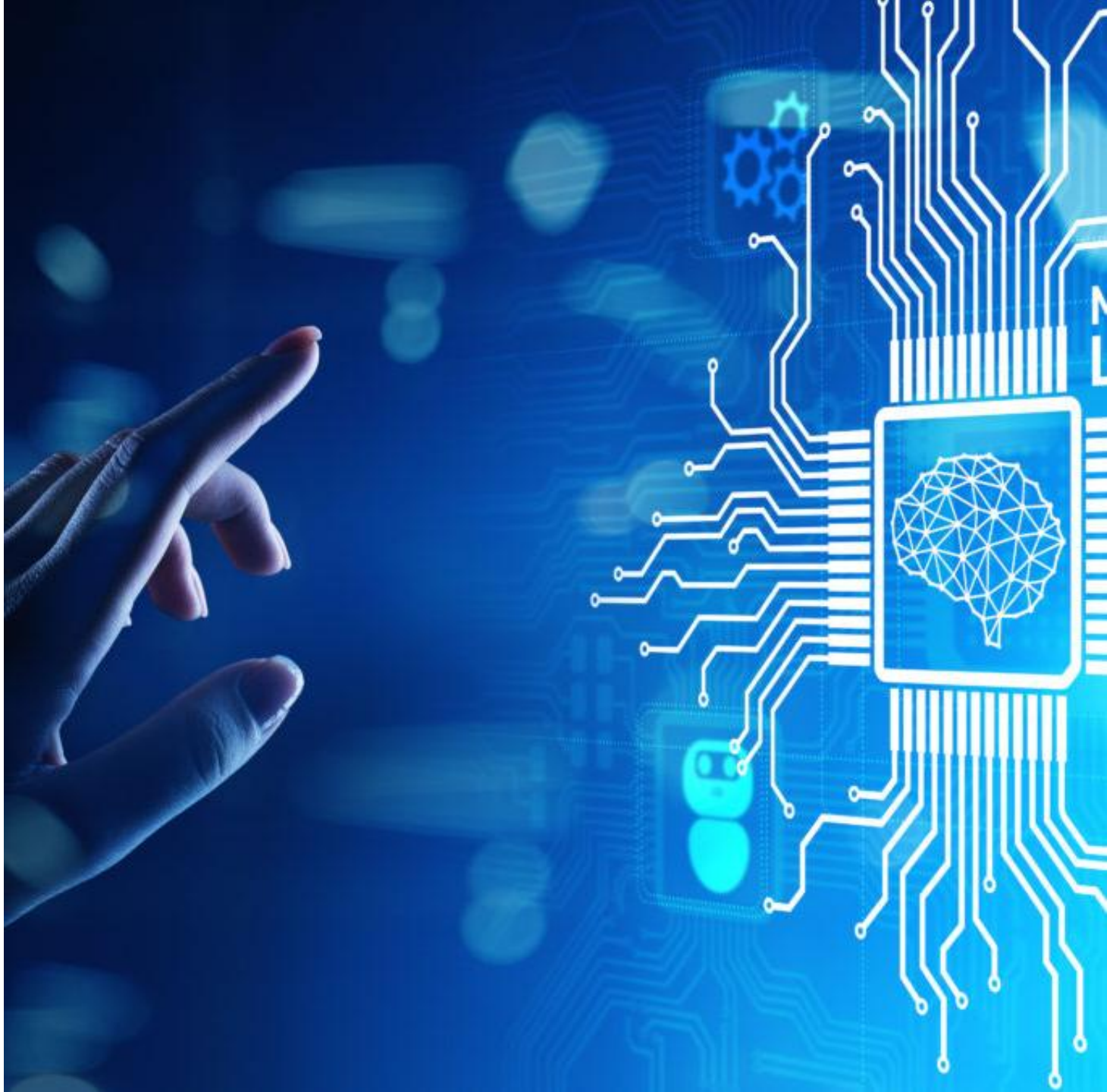
DIGITAL LEARNING HUB

easi

Introduction to Machine Learning

Zeyneb GASMI
Data Science & Business Intelligence Consultant



Agenda: Day 2

9 AM – 12 PM:

1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5: Evaluate your Models

Agenda: Day 2

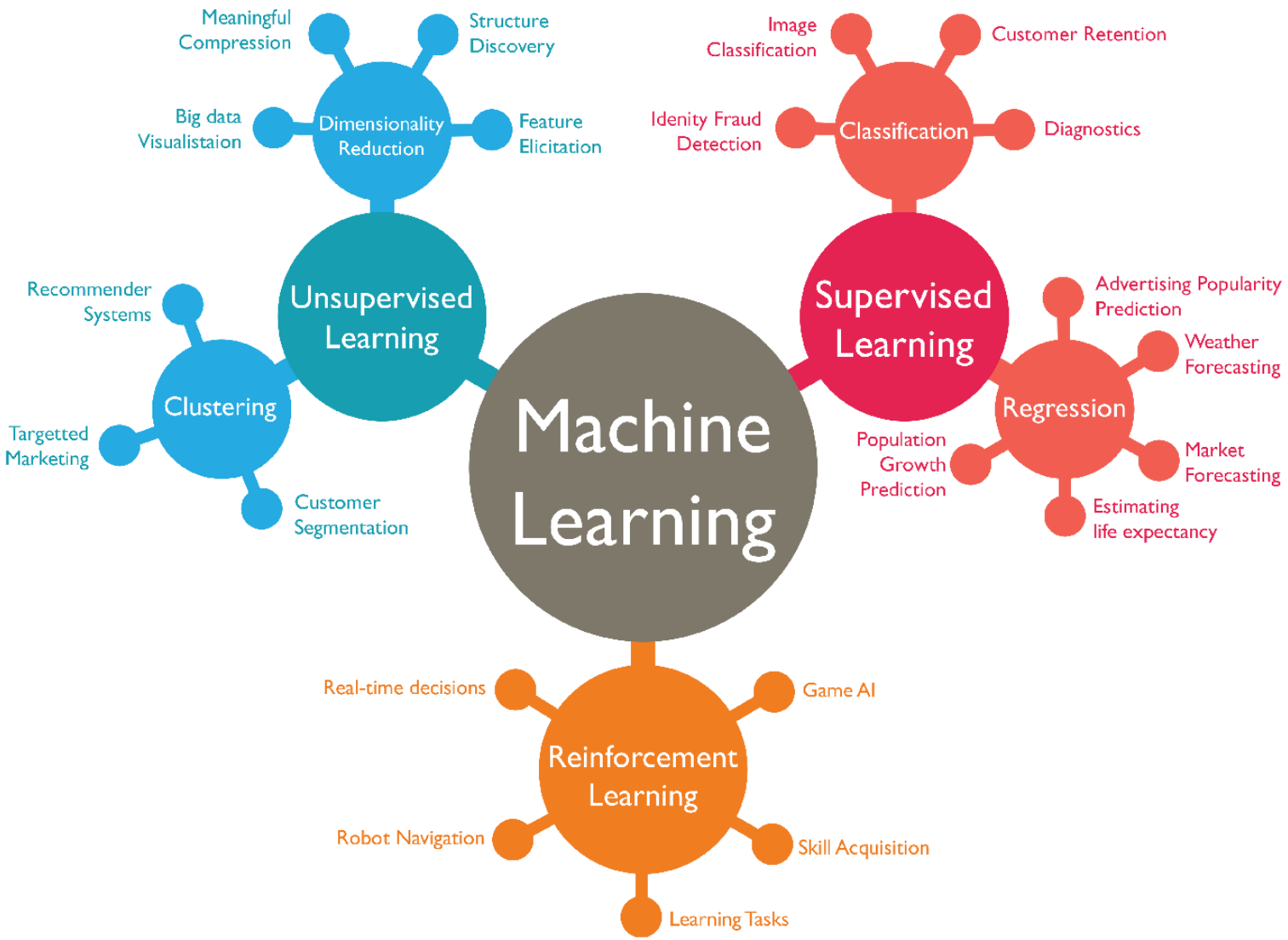
9 AM – 12 PM:

1. **Supervised Machine Learning : Reminder**
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

Machine Learning Types



Machine Learning Types : Supervised Learning

Types de Machine Learning

Supervised

✓ Using labeled data to train algorithms.

Examples of techniques: Classification, Regression.

Unsupervised

✓ Using unlabeled data to discover structures or groupings.

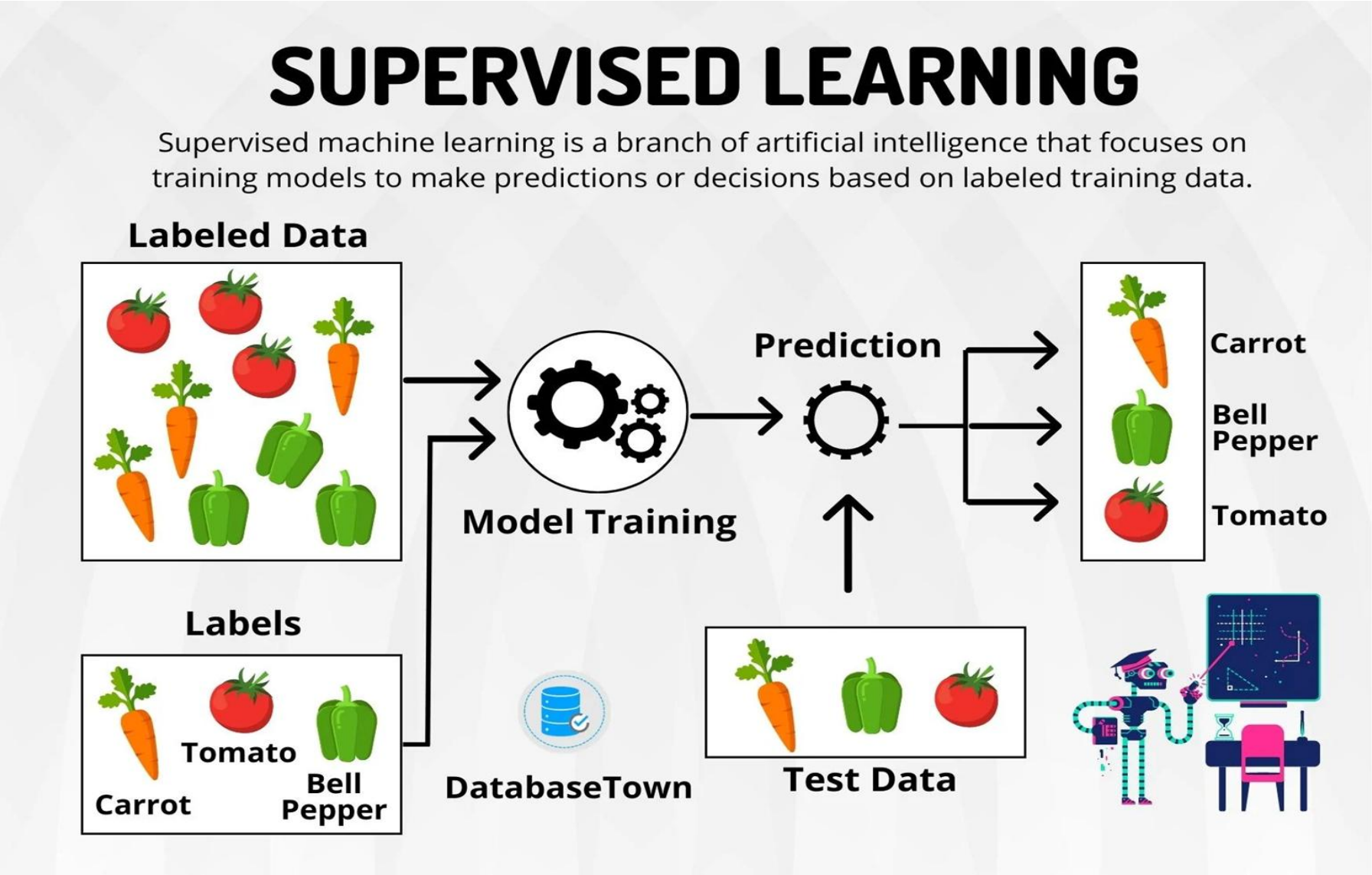
Examples of techniques: Clustering, Dimensionality reduction

Reinforcement learning

✓Learning by interaction with an environment, based on rewards.

Example: Optimization of strategies in games

Supervised Learning : How it works ?



Agenda: Day 2

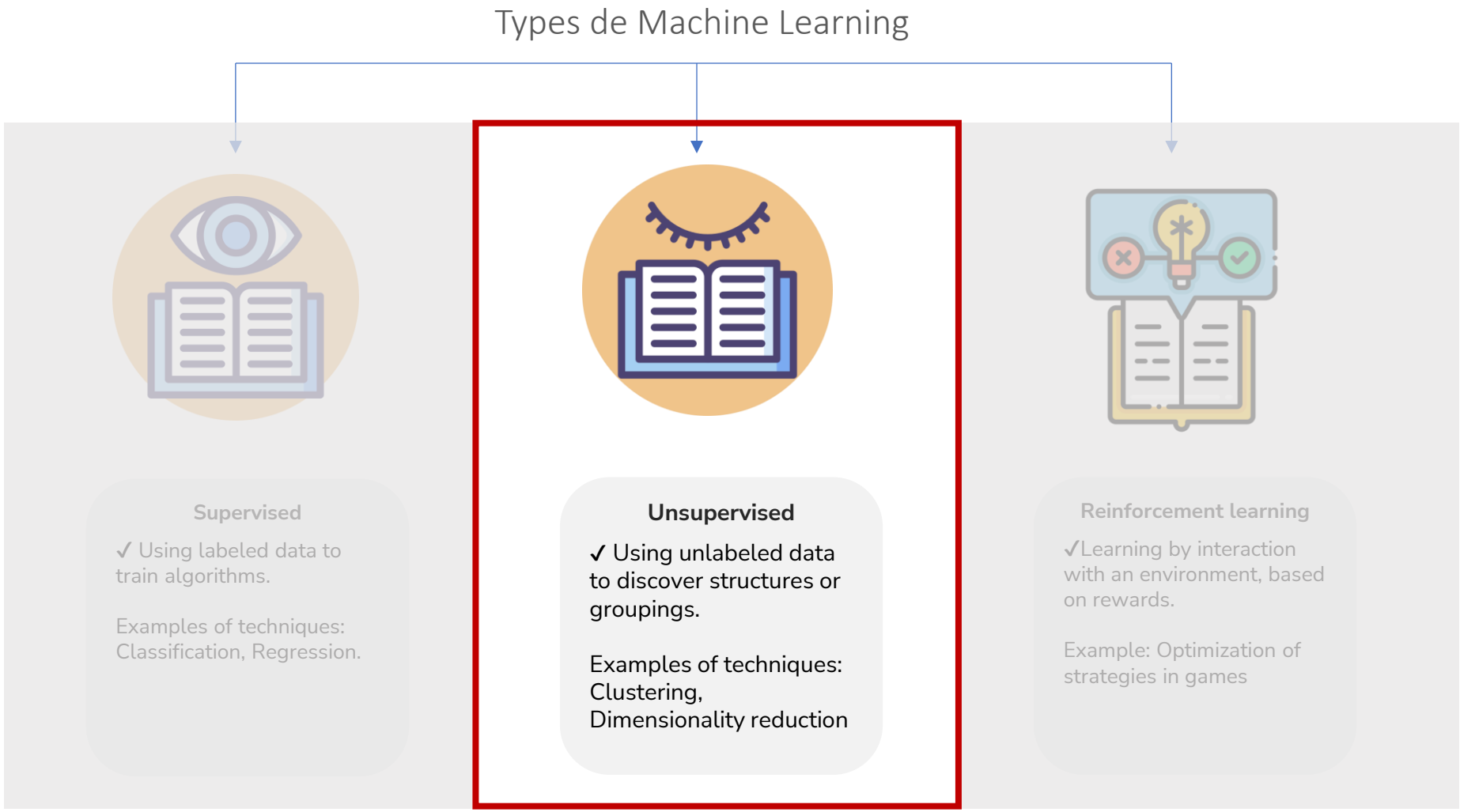
9 AM – 12 PM:

1. Supervised Machine Learning : Reminder
2. **Unsupervised Machine Learning**
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

Machine Learning Types : Unsupervised Learning



Unsupervised Learning : unlabelled Dataset

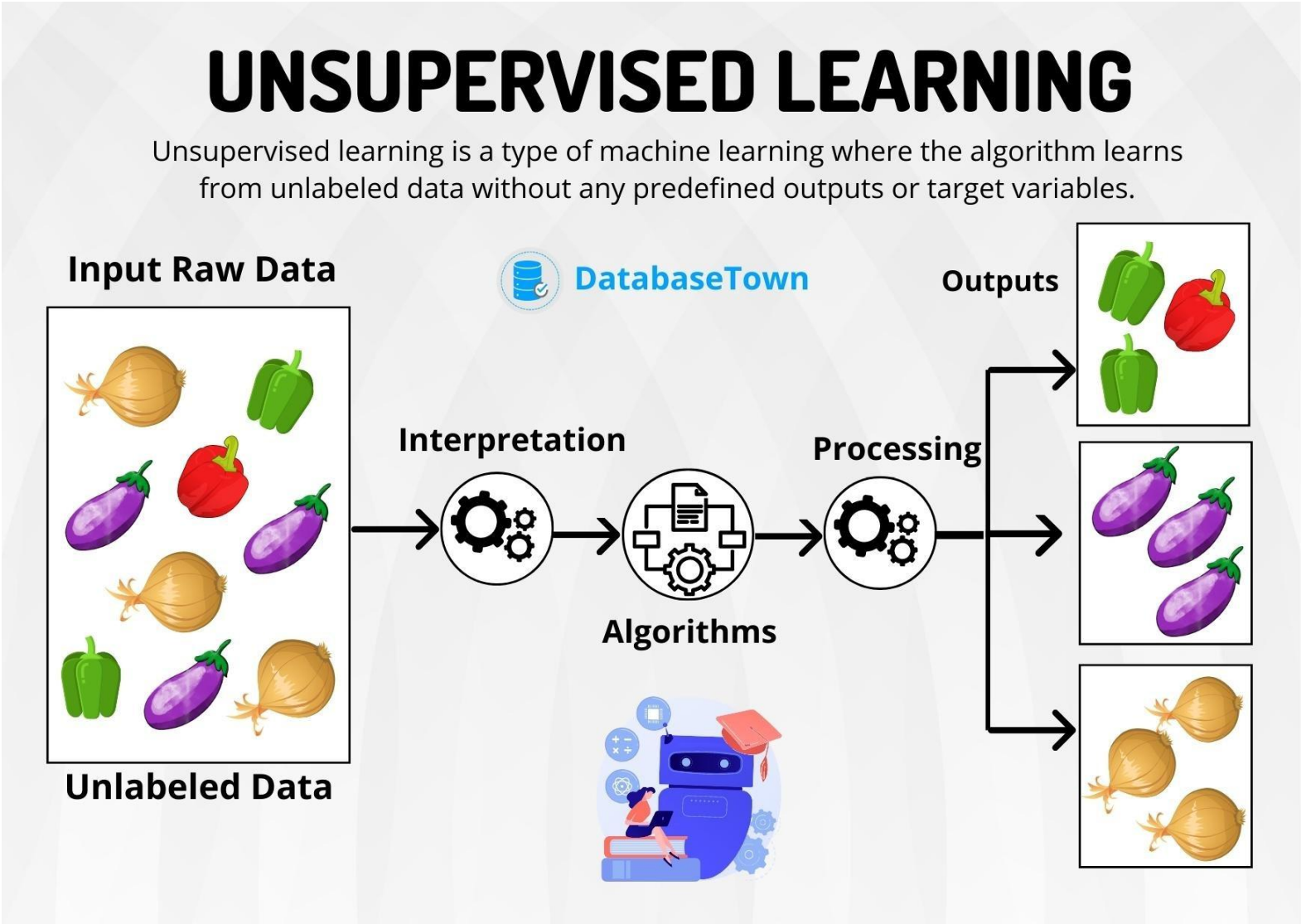
- **Unlabelled Data :** A dataset with only features with **no target to predict.**

Features X

Target Y

Features					Label
Position	Experience	Skill	Country	City	Salary ($)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	1049
Developer	2	1	USA	New York	
Developer	3	1	USA	New York	
Developer	4	1	USA	New York	1104
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Unsupervised Learning : How it works ?



Unsupervised Learning : Clustering

• **Clustering :** This is the task of **grouping our data** into clusters based on similarity.

Before Clustering :




Unsupervised Learning : Clustering

• **Clustering :** This is the task of **grouping our data** into clusters based on similarity.

After Clustering :

Group A

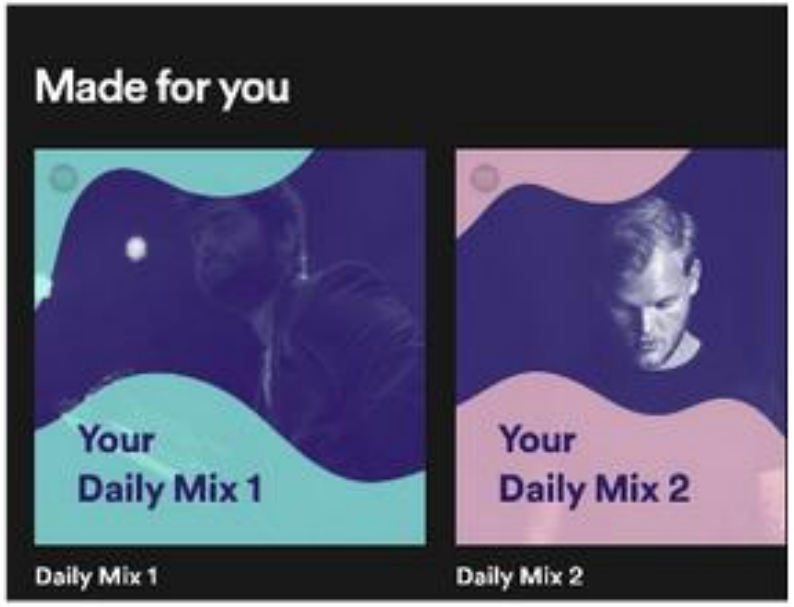
Group B

Group C

Group C

Unsupervised Learning : Clustering - Examples

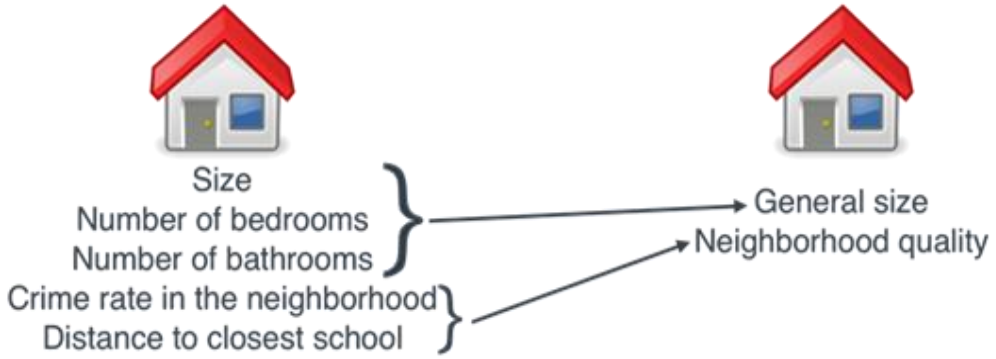
Unsupervised Learning : Clustering - Examples



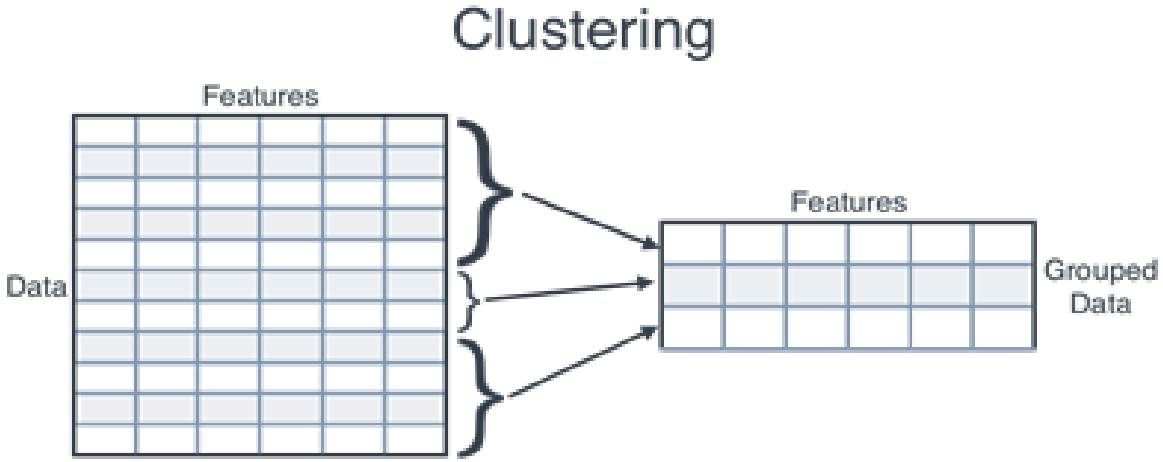
Clustering can also be used in recommendation engines. Let's say you want to recommend songs to your friends. You can look at the songs liked by that person and then use clustering to find similar songs and finally recommend the most similar songs.

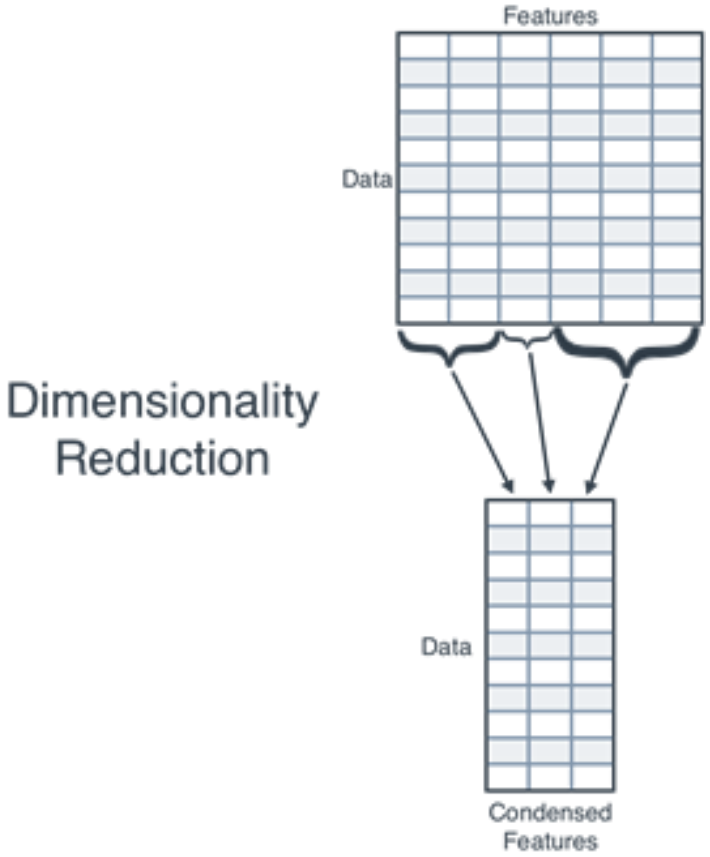
Unsupervised Learning : Dimentionality Reduction

• **Dimensionality reduction :** A very useful preprocessing statistical tool that **simplifies** data without losing –much- information . IT converts a high-dimensional dataset to a low-dimensional one.



Unsupervised Learning : Clustering Vs Dim Reduction





Unsupervised Learning : Association

- **Association :** Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an itemset occurs in a transaction.

- **Examples :**

Medicine.

Retail.

User experience (UX) design.

Entertainment.

Unsupervised Learning : Association - Use Case

Beer and Diapers Tale

Agenda: Day 2

9 AM – 12 PM:

1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. **K-Means**
4. Lab 3: Build your first unsupervised learning model

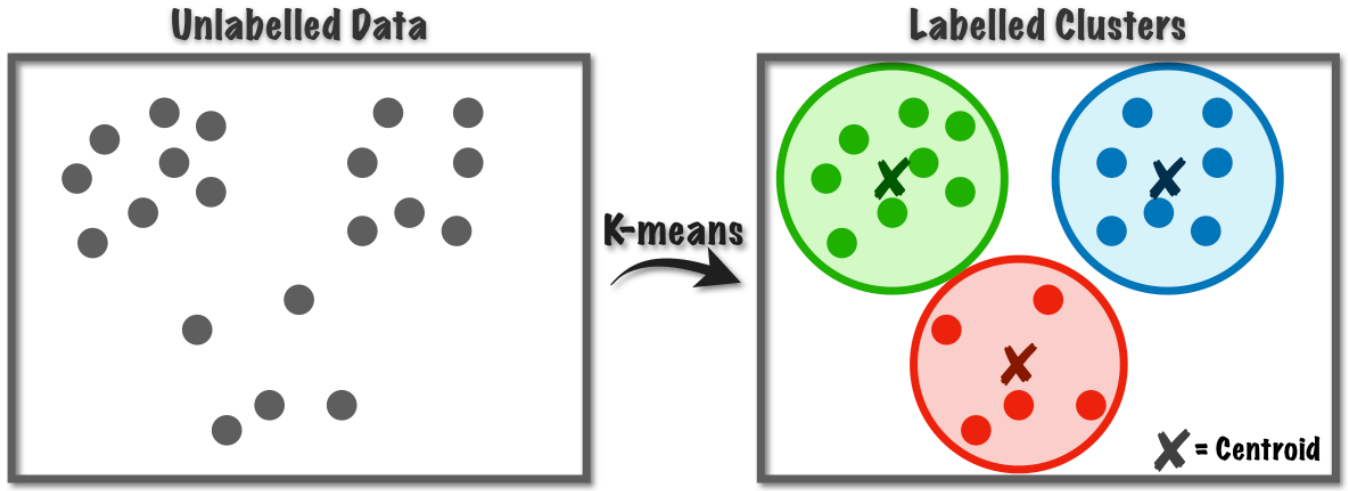
1 PM – 4 30 PM:

1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

K-Means : A Clustering Algorithm

K-means : is a clustering algorithm. It is an iterative algorithm that tries to partition the dataset into *K* pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**.

It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.



K-Means : Domains of application

Customer Segmentation

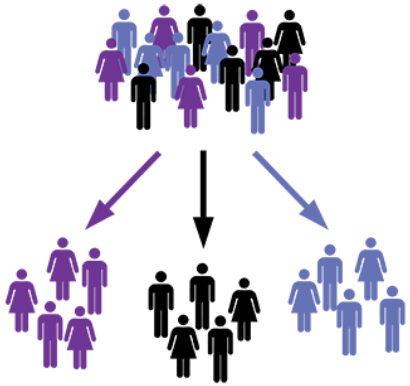
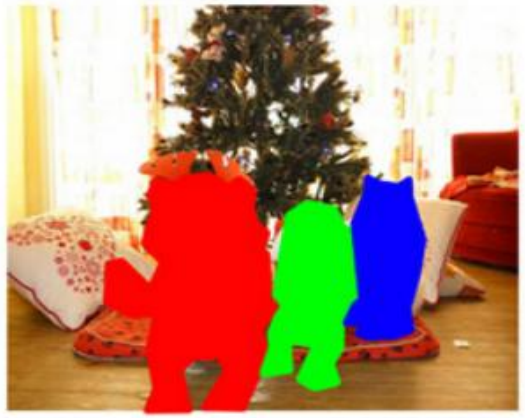
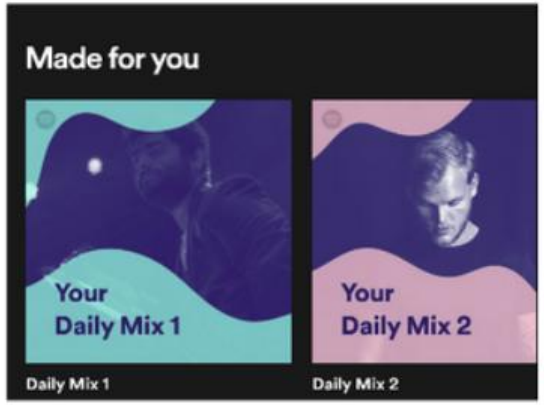


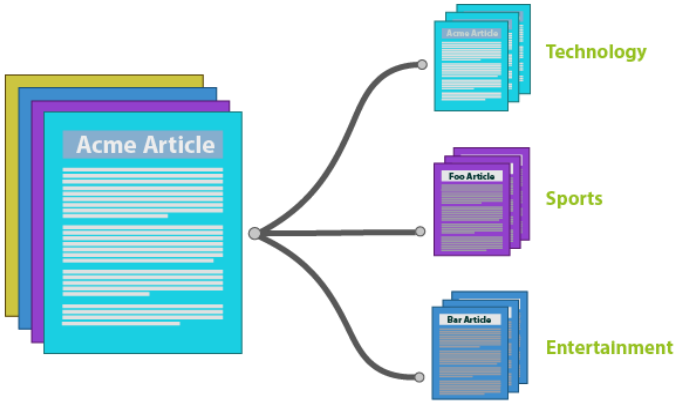
Image segmentation



Recommandation Engines



Document clustering



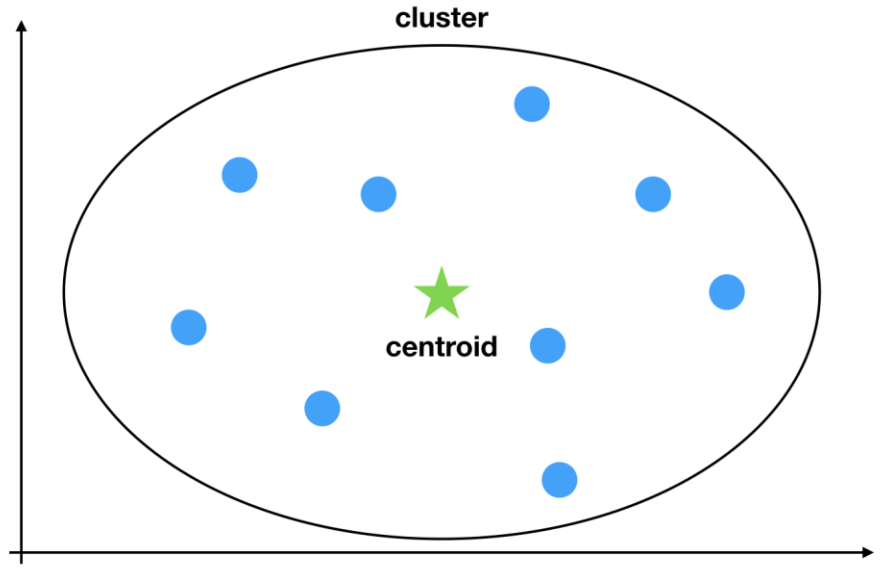
K-Means : Key concepts

Cluster :

Collection of data points grouped together based on similarity. It represents a subset of the dataset that shares common characteristics.

Centroid :

The representative point of a cluster is called a centroid. It is the center of the samples that belong to the cluster and works as a prototype of the cluster. Finding the appropriate centroids that partition samples in a good manner is the goal of the K-means algorithm.



K-Means : Key concepts

Euclidean distance

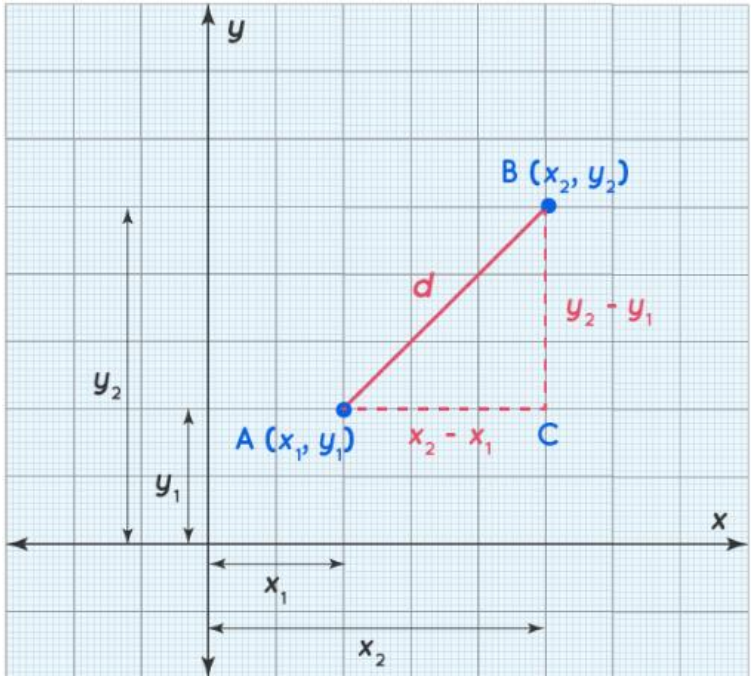
In coordinate geometry, Euclidean distance is the distance between two points. To find the two points on a plane, the length of a segment connecting the two points is measured.

Formula

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Application

$$d(A,B) = \sqrt{[(5-2)^2+(5-2)^2]} = 4{,}24$$



A(2,2) and B(5,5)

K-Means : Algorithm

The goal of the k-means algorithm is to partition a dataset into k clusters by minimizing the within-cluster variance.

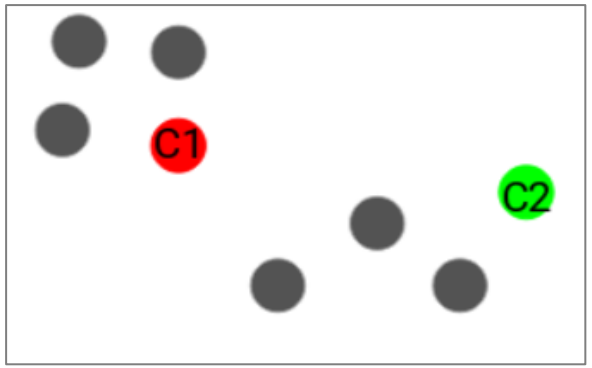
The steps are :

1. Choose a random number of clusters k (for example k=2)

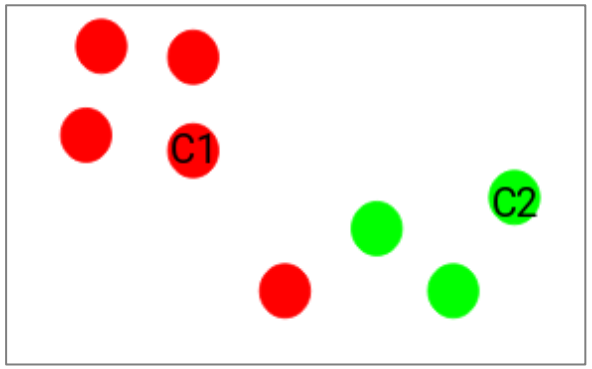
2. Select k random points from the data as centroids, the center of the cluster (here we have C1 and C2 as centroids)

3. Select randomly the centroid for each cluster. (C1 for cluster 1 and C2 for cluster 2)

4. Assign all the points to the closest centroid. To do this, we need to calculate the euclidean distance between the point and the centroid by using the euclidean distance formula



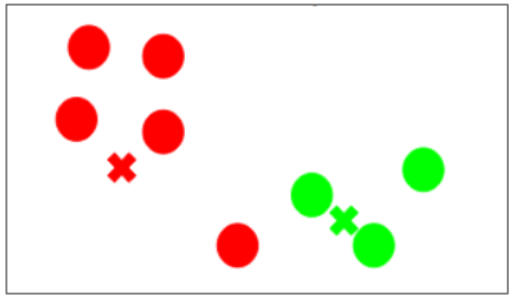
Step 2



Step 4

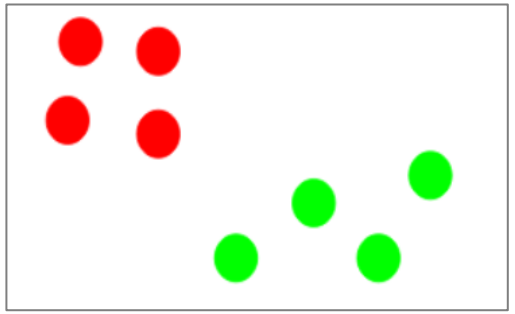
K-means : Algorithm

5. Recompute the centroids of newly formed clusters



Step 5

6. Repeat steps 3 and 4

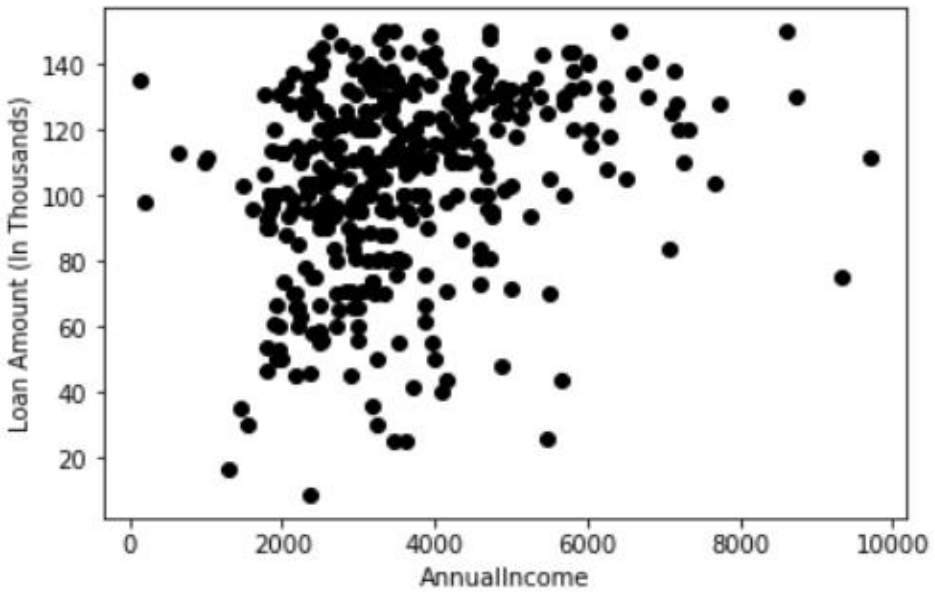


Step 6

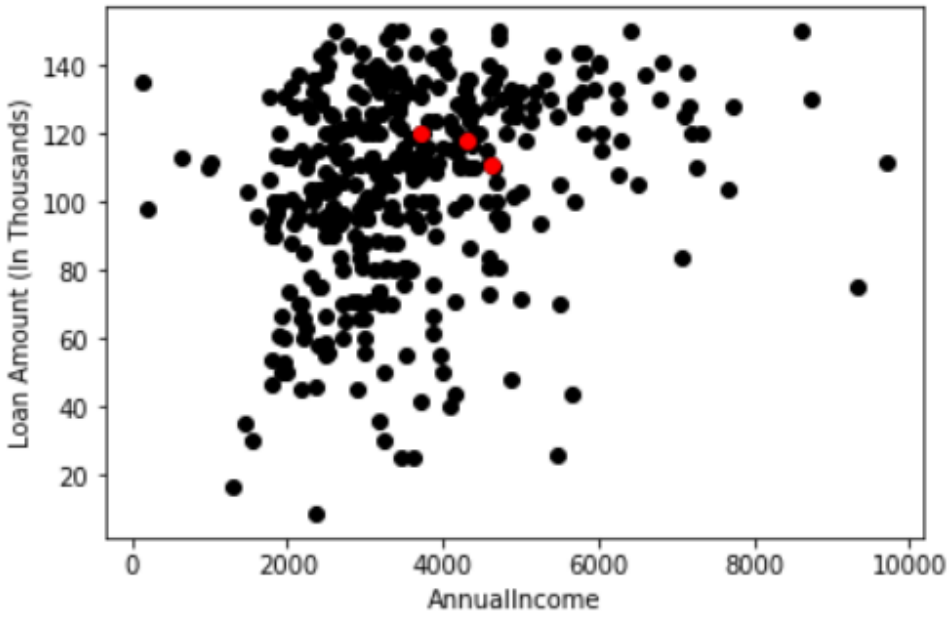
7. One we have one of this criteria, we stop the algorithm:
   1. Centroids of newly formed clusters do not change
   2. Points remain in the same cluster
   3. Maximum number of iterations is reached

K-Means : Application example

1. Intial Dataset

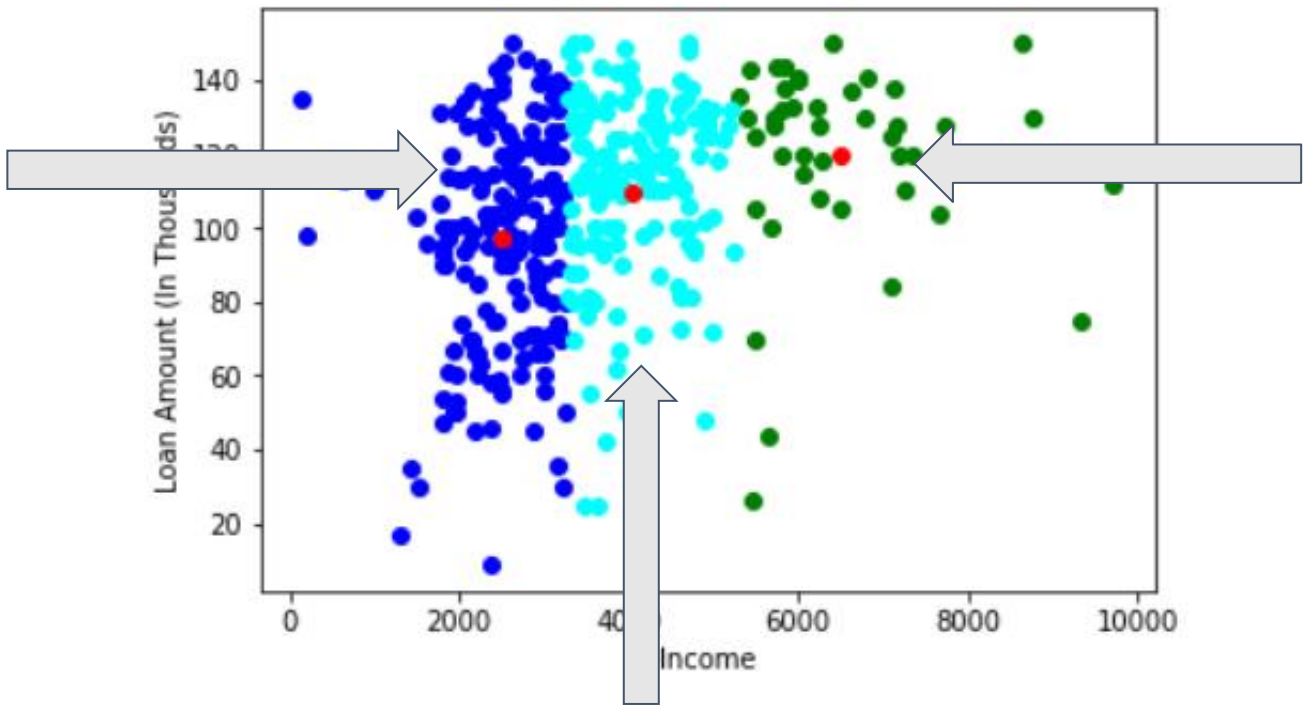


2. The algorithm selectes 3 random points from the dataset as centroids (the points in red)



K-Means : Application example

3. Final Result of the algorithm :



NB : The algorithm may take lot of iterations until arriving to the stopping criteria.

K-Means : Optimize with Elbow method

To optimise the number of the clusters, the most commonly used method is Elbow method:

A) Execute the K-means algorithm for a range of cluster numbers, typically starting from 1 up to a certain maximum value.

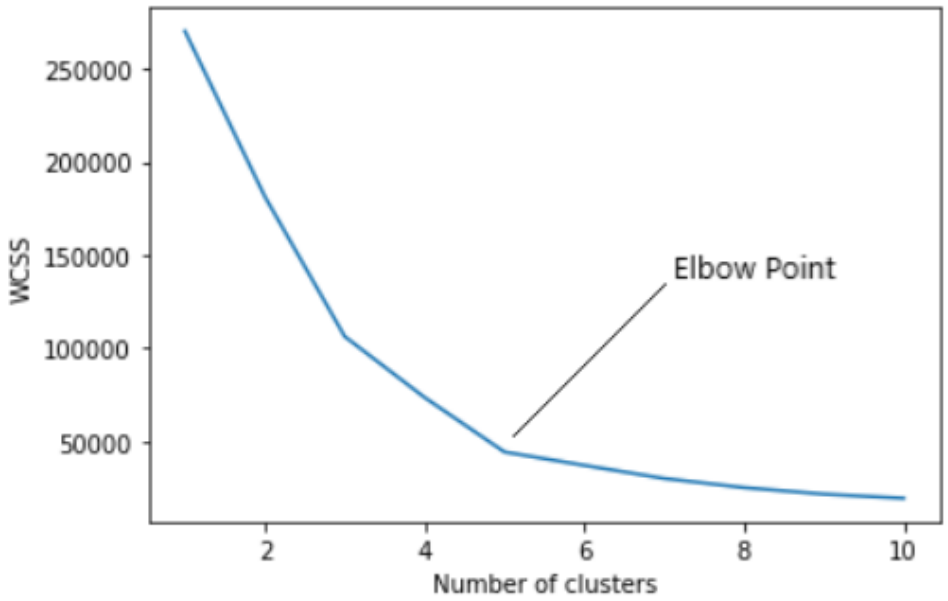
B) For each clustering iteration we compute the Within-Cluster Sum of Squares (WCSS)

$$WCSS = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{i,k} ||\mathbf{x}_i - \mathbf{c}_k||^2$$

C) Plot the WCSS against the number of clusters: Plot the WCSS values against the number of clusters

K-Means : Optimize with Elbow method

D) Identify the "elbow" point by examining the plot and identify the point where the rate of decrease in WCSS starts to slow down, forming an "elbow" shape. This point indicates the optimal number of clusters.



E) Choose the number of clusters corresponding to the "elbow" point as the optimal number for your dataset, in the image above the optimal number of clusters is 5.

Agenda: Day 2

9 AM – 12 PM:

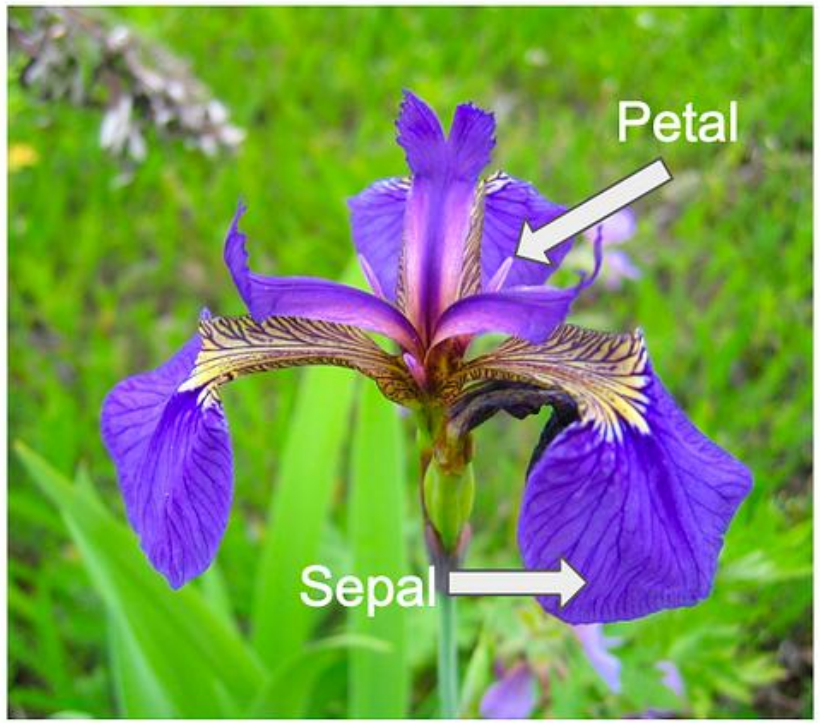
1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. **Lab 3: Build your first unsupervised learning model**

1 PM – 4 30 PM:

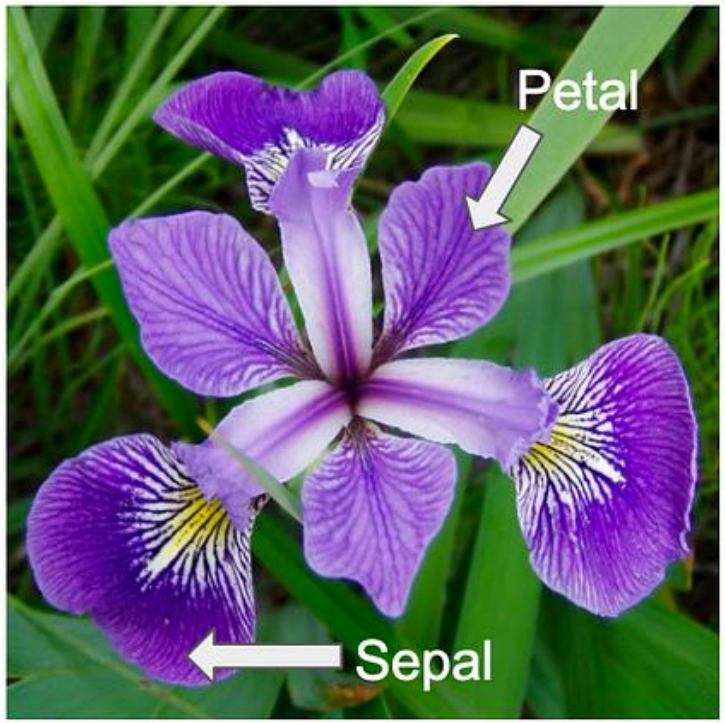
1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

Lab 3 : Build your 1st Unsupervised Model with K-Means

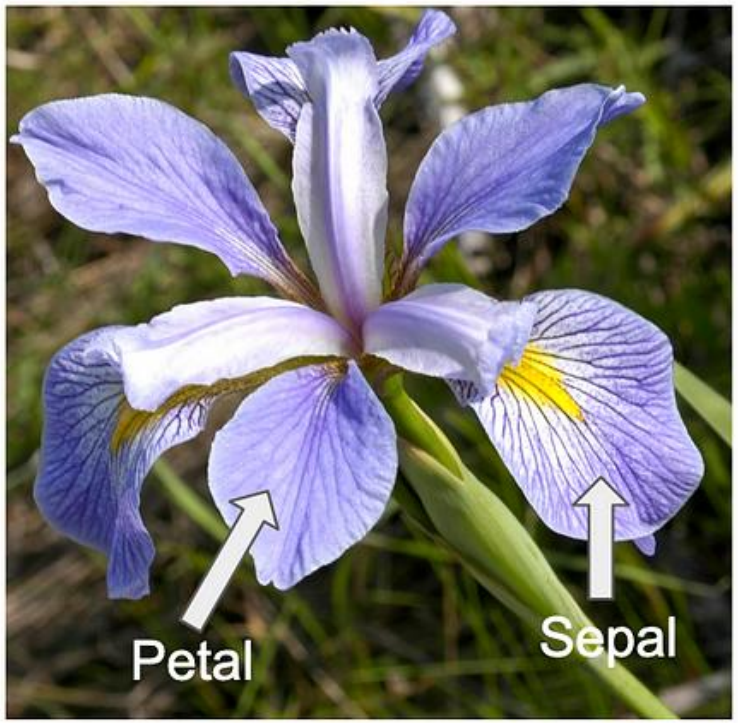
Iris setosa



Iris versicolor



Iris virginica



Agenda: Day 2

9 AM – 12 PM:

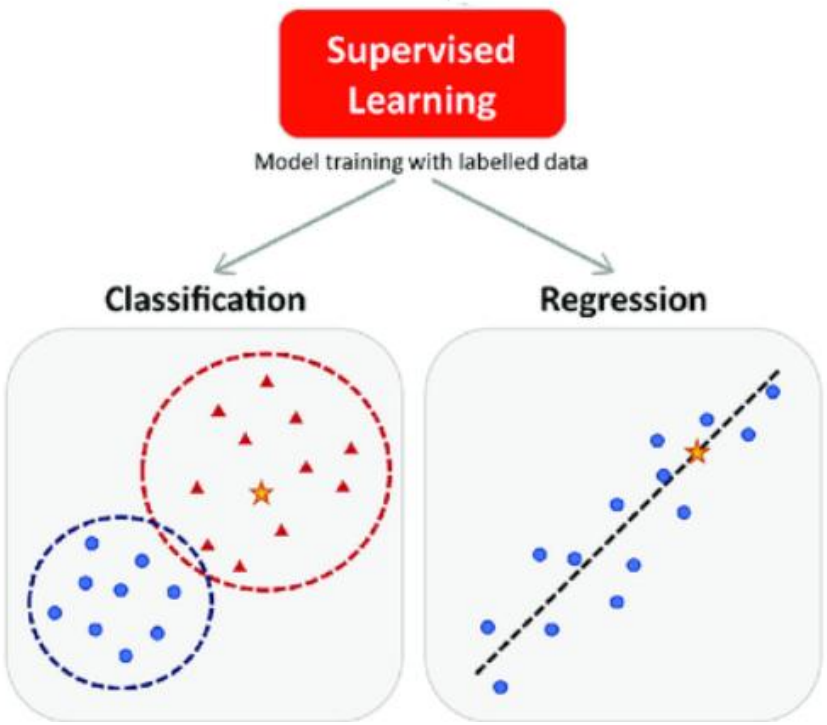
1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. **K Nearest Neighbors**
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

K Nearest Neighbors : Definition

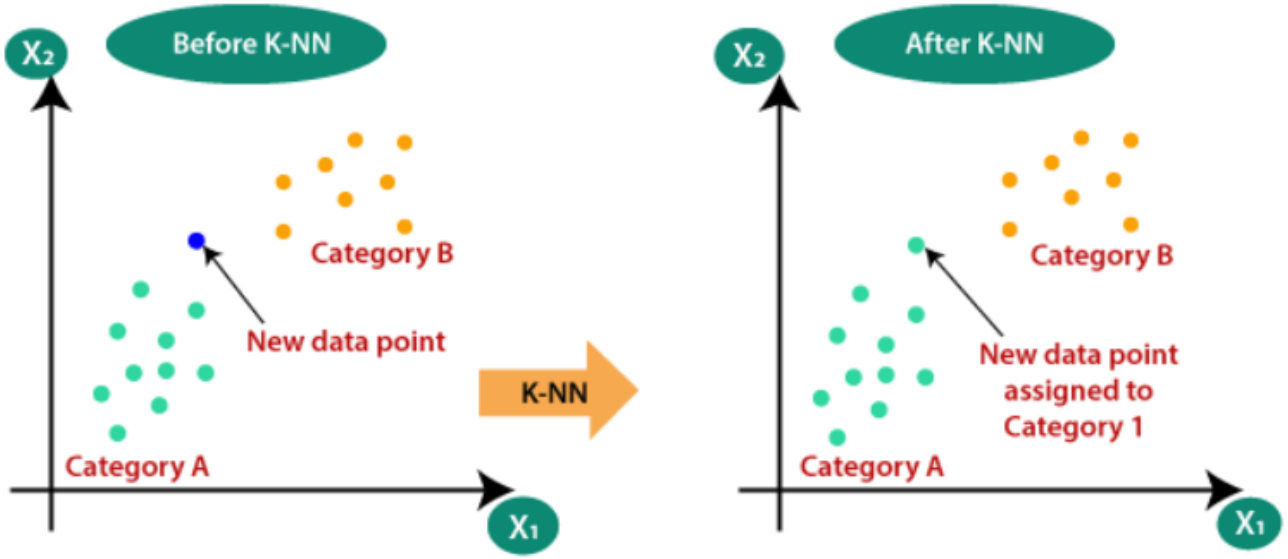
- **K-Nearest Neighbours (KNN)** algorithm is a popular and simple machine learning technique used for classification and regression tasks.
- It relies on the idea that similar data points tend to have similar labels or values.
- It stores all the available data and when a new data point appears it can be easily classified based on similarities.



K Nearest Neighbour

K Nearest Neighbors : Application

To classify a new data point into one of the two existing categories (A or B), we use the KNN algorithm. Based on the spatial proximity of points, KNN assigns to the new point the category that contain the most of nearest neighbours .



Application Example

K Nearest Neighbors : Instance based learning

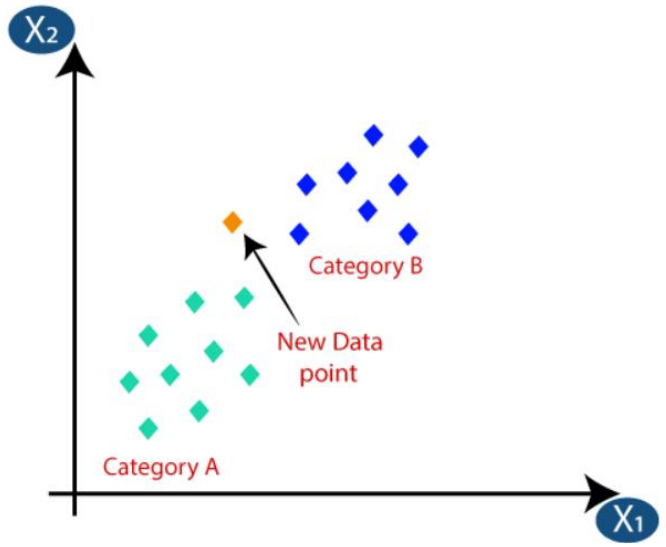
- Instance-based learning are the systems that **learn the training examples by heart** and then **generalizes to new instances** based on some similarity measure.
- It is also known as **memory-based learning** or **lazy-learning.**

Advantages:

- Instead of estimating for the entire instance set, local approximations can be made to the target function.
- Adapt to new data easily.

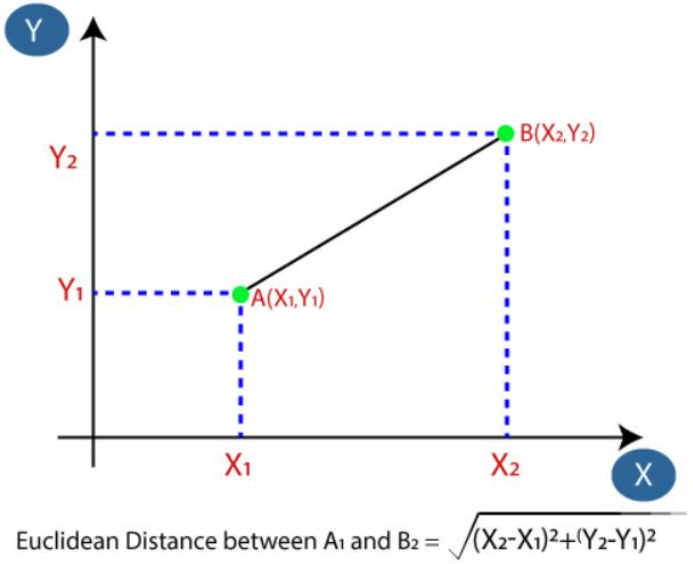
K Nearest Neighbors : Algorithm

1. We select the number K of the neighbors, for example 5.



Calculate distances

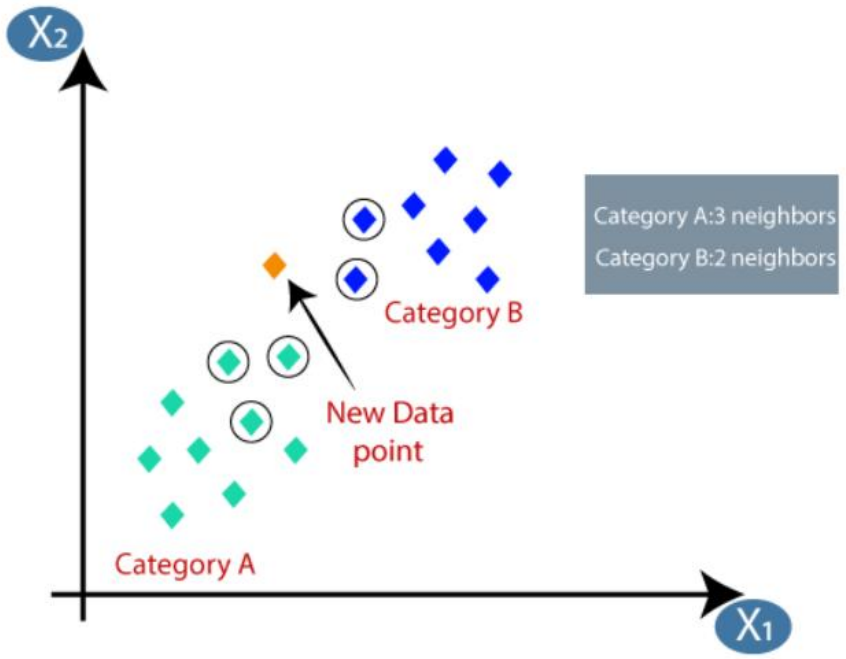
2. We Calculate the Euclidean distance between the new data point and all the points in the dataset.



K Nearest Neighbors : Algorithm

Identify the K nearest neighbors

3. Take the K (in our case 5) nearest neighbors as per the calculated Euclidean distance. (we take the 5 nearest points to the new data point)



Voting mechanism

4. Among these k neighbors, count the number of the data points in each category. (3 in category A and 2 in category B)

5. Assign the new data points to that category for which the number of the neighbor is maximum. (Category A because it contains 3 nearest neighbors)

K Nearest Neighbors : Distance metrics and Feature space

- ✓ Data is in feature space, KNN assumes that the distance between the data points in feature space can be evaluated using various distance metrics like Euclidean, Manhattan, cosine similarity etc.

- ✓ The most used distance metrics in KNN algorithm are:

Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Cosine similarity

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

Manhattan Distance

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

K means vs K nearest Neighbors

Model	**K means**	**K nearest Neighbor**
Learning	Unsupervised	Supervised
Algorithm type	Clustering	Classification or Regression
Purpose	Clustering data point into k clusters	Classifying or regressing based on k nearest neighbor
Input	Unlabeled data	Labeled data
Output	Produces K cluster centroids representing cluster centers	Assigns a class label or regression value based on the majority class or average value of K nearest neighbors

Agenda: Day 2

9 AM – 12 PM:

1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. K Nearest Neighbors
2. **Lab 4: Use K-Nearest Neighbors to solve a classification problem**
3. Evaluation Metrics
4. Lab 5 : Evaluate your Models

Lab 4: Use K-Nearest Neighbors to solve a classification problem

In this lab, we will be using the “breast cancer dataset”. it contains medical records of breast tumor characteristics, with features like radius, texture, and concavity. Each record is labeled as benign or malignant

Your task is to build a K-Nearest Neighbors Model to Classify Tumors as « Benign » or « Malignant »

Agenda: Day 2

9 AM – 12 PM:

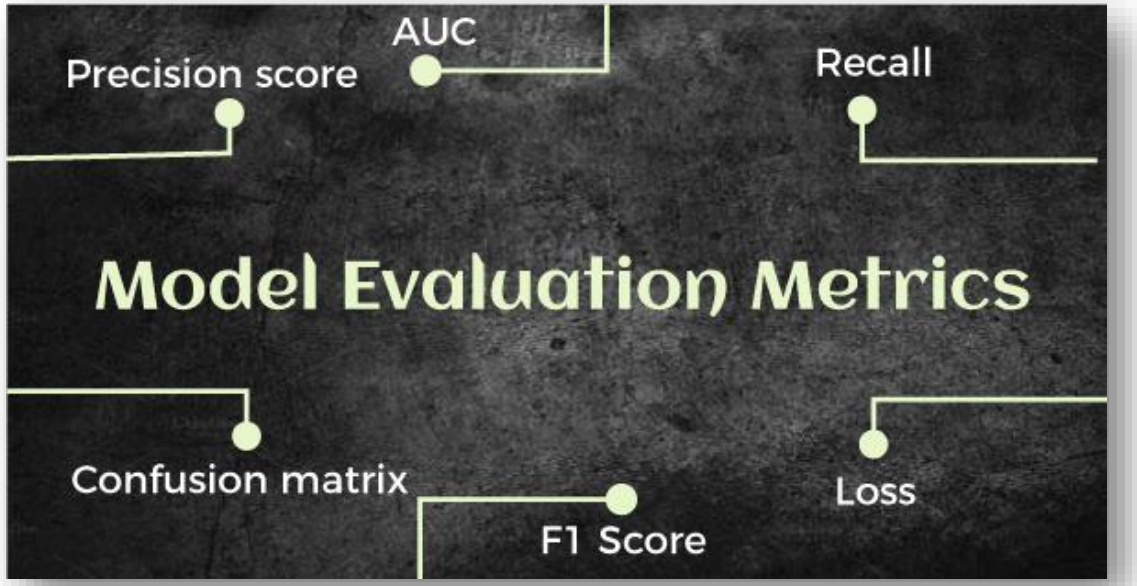
1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

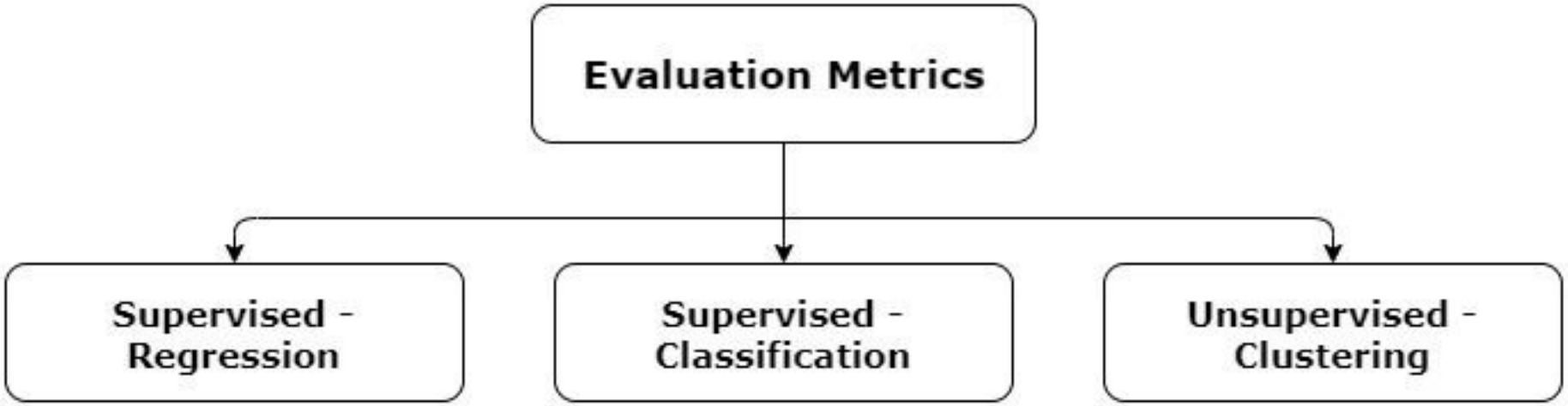
1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. **Evaluation Metrics**
4. Lab 5 : Evaluate your Models

Evaluation Metrics : Why to use ?

- **Measure Performance:** Evaluation metrics help to measure how well a machine learning model is performing on a given task.
- **Compare Models:** They allow comparison between different models or different configurations of the same model to find the best performing one.
- **Identify Problems:** Metrics can reveal if a model is underfitting, overfitting, or if there are issues with data imbalance.
- **Guide Improvements:** They provide feedback on what aspects of the model need improvement, guiding further development and tuning.



Evaluation Metrics : Types



1. Mean Absolute Error(MAE)
2. Mean Squared Error(MSE)
3. Root Mean Squared Error (RMSE)
4. R squared
5. Adjusted R-Squared

1. Accuracy
2. Precision
3. Recall
4. F1 Score
5. Specificity

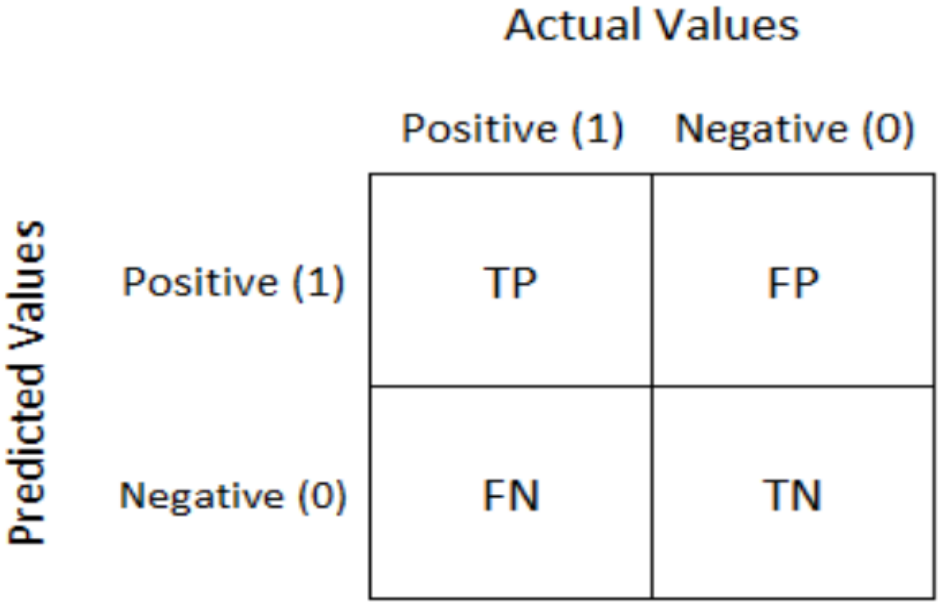
1. Rand Index
2. Adjusted Rand Index

Classification metrics : Confusion matrix

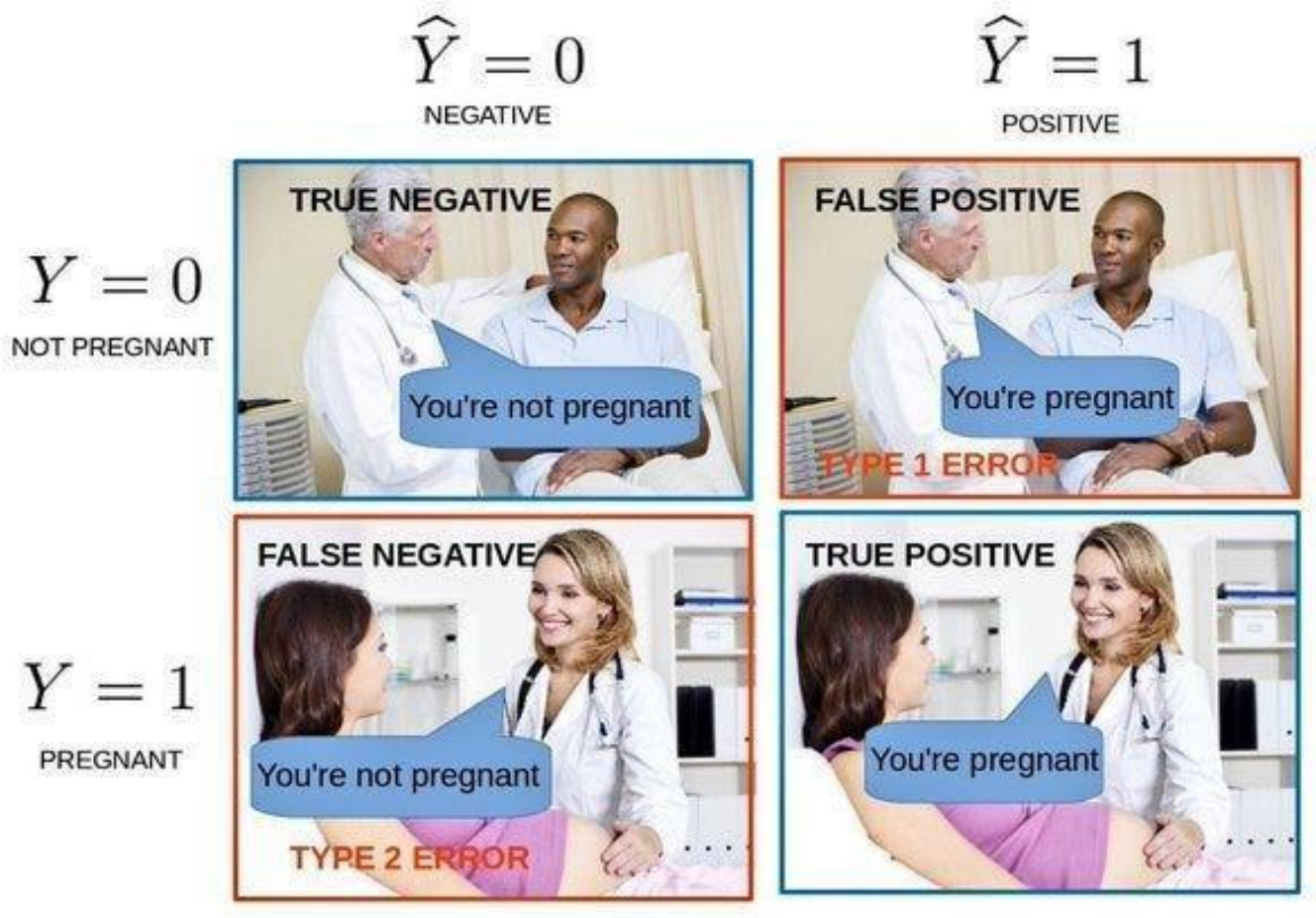
- ✓ **Confusion Matrix** is a table used to evaluate the performance of a classification model. It shows the counts of actual Vs predicted classifications, providing a detailed breakdown of the model's performance.

- ✓ The confusion matrix is typically structured as follows for a binary classification problem:

  - ✓ **True Positive (TP):** The number of instances where the model correctly predicted the positive class.
  - ✓ **False Positive (FP):** The number of instances where the model incorrectly predicted the positive class (Type I error).
  - ✓ **True Negative (TN):** The number of instances where the model correctly predicted the negative class.
  - ✓ **False Negative (FN):** The number of instances where the model incorrectly predicted the negative class (Type II error).



Classification metrics : Confusion matrix - Example



Classification metrics : Accuracy & Precision

Accuracy : the proportion of true results (both true positives and true negatives) among the total number of cases examined.

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

Precision : the proportion of true positive results in the predicted positive cases.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Classification metrics : Recall & F1-Score

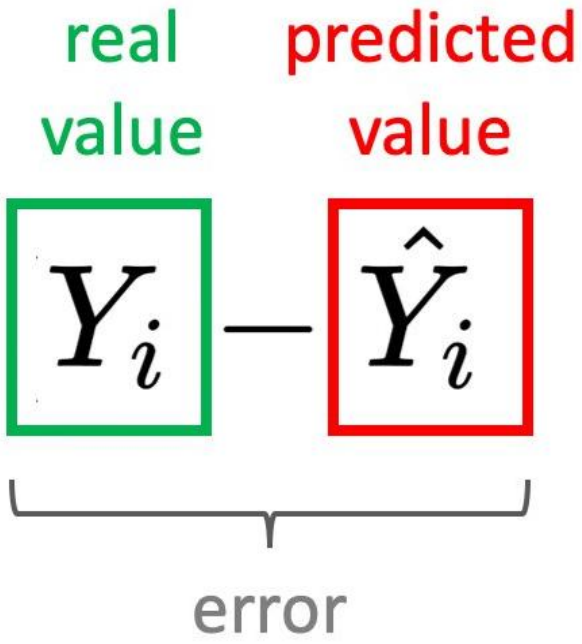
Recall : (Also called Sensitivity) is the proportion of true positive results in the actual positive cases.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

F1-Score : The harmonic mean of precision and recall, providing a single metric that balances both concerns.

$$F1 = 2. \frac{Precision \times Recall}{Precision + Recall}$$

Regression metrics : Assumption



Regression metric : MSE

MAE – Mean Absolute Error :

- ✓ The average of the absolute differences between the predicted values and the actual values.
- ✓ It provides a straightforward interpretation of the average error magnitude in the same units as the target variable

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

Regression metrics : MSE & MAE

MAE – Mean Absolute Error :

- ✓ The average of the absolute differences between the predicted values and the actual values.
- ✓ It provides a straightforward interpretation of the average error magnitude in the same units as the target variable

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

MSE – Mean Squared Error :

- ✓ The average of the squared differences between the predicted values and the actual values
- ✓ It penalizes larger errors more than smaller ones, making it sensitive to outliers.

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

Regression metrics : RMSE & R2

RMSE – Root Mean Squared Error :

- ✓ Computes the square root of the Mean Squared Error.
- ✓ It provides a measure of error in the same units as the target variable and is easier to interpret than MSE.

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

R2 – R Squared :

- ✓ Represents the proportion of the variance in the dependent variable that is predictable from the independent variables.
- ✓ It indicates how well the data fit the regression model, with values closer to 1 indicating a better fit.

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

where $\bar{y}$ is the mean of the actual values.

Agenda: Day 2

9 AM – 12 PM:

1. Supervised Machine Learning : Reminder
2. Unsupervised Machine Learning
3. K-Means
4. Lab 3: Build your first unsupervised learning model

1 PM – 4 30 PM:

1. K Nearest Neighbors
2. Lab 4: Use K-Nearest Neighbors to solve a classification problem
3. Evaluation Metrics
4. **Lab 5 : Evaluate your Models**

Lab 5 : Evaluate your Models

In this lab, You are asked to use Evaluation Metrics available in Sklearn Library to Evaluate all your models created in your previous labs

DIGITAL LEARNING HUB

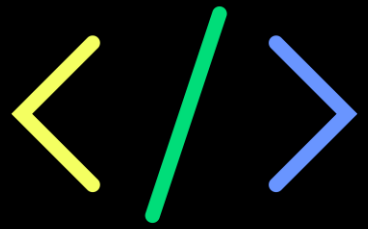
Make IT real

Design thinking

Blockchain

Coding

Cybersecurity



Please give us feedback about the course

https://www.dlh.lu/survey/start/979e52d3-6032-41cd-bf6a-b192bea9f510

Thank you!

www.kozalys.com

www.easi.net

z.gasmi@easi.net