

INTRO

PIPELINE

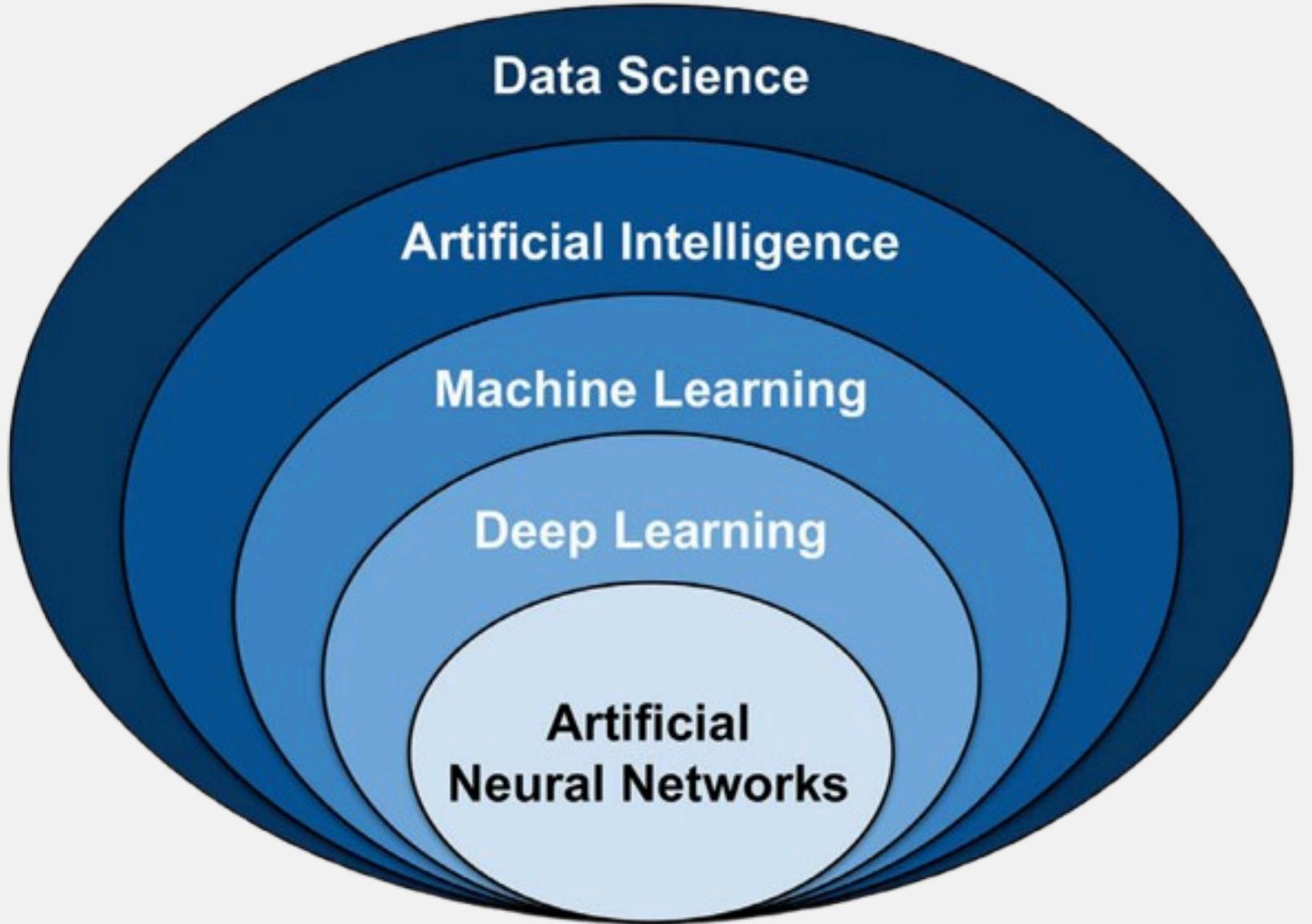
TYPES

OTHERS

Topic 1

Machine Learning





ML is a **subset of AI** that uses algorithms to learn from data and make predictions without hardcoded rules.

So instead of writing specific instructions, we feed data to the model and let it **learn by itself by detecting patterns**.

“free money”

“limited time”

“you got iPhone”

“exclusive deal”

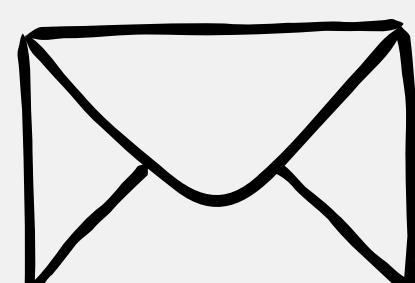
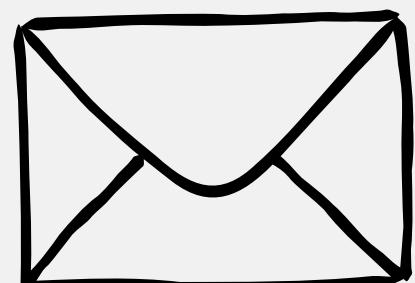
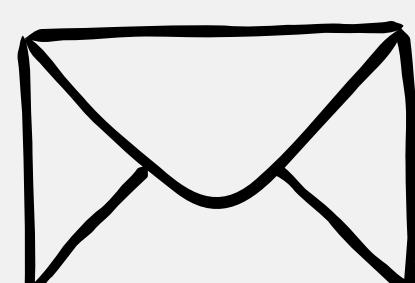
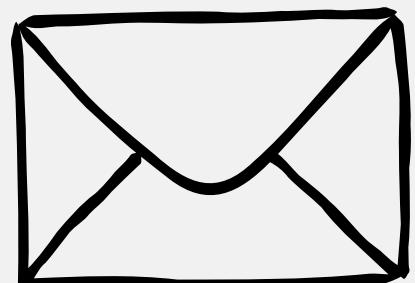
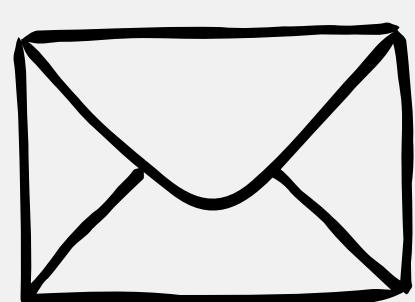
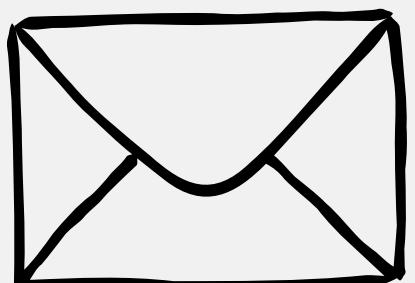
“100% guaranteed”

“giveaway”

“last chance”

“win lucky draw”

MARK EMAILS WITH
THESE WORDS AS
SPAM!



Why is ML important?

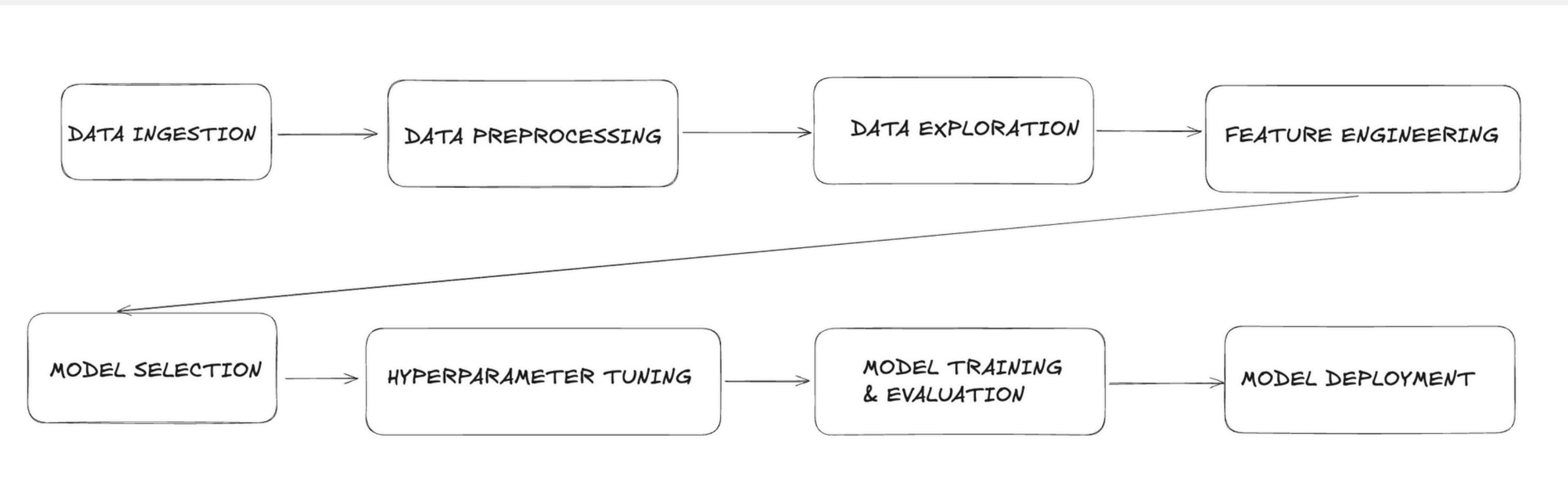


- **Automation:** Automate complex decision-making processes
- **Pattern Recognition:** Find hidden patterns in large datasets
- **Prediction:** Make predictions about future events
- **Personalization:** Customize experiences (like Netflix recommendations)



ML Pipeline

Steps to build an ML model:

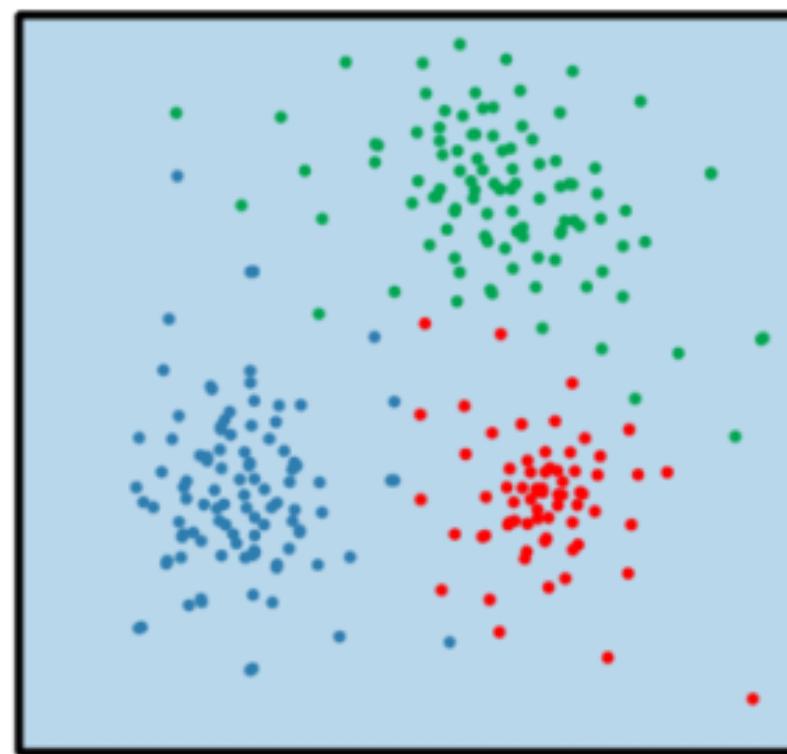


Step	Short Definition
1. Data Ingestion	Load data from files, APIs, or databases
2. Data Preprocessing	Clean and fix messy data
3. Data Exploration	Look at data to understand patterns
4. Feature Engineering	Create better input variables
5. Model Selection	Pick the best algorithm
6. Hyperparameter Tuning	Adjust settings for better performance
7. Model Training and Evaluation	Train model and test accuracy
8. Model Deployment	Put model into real use

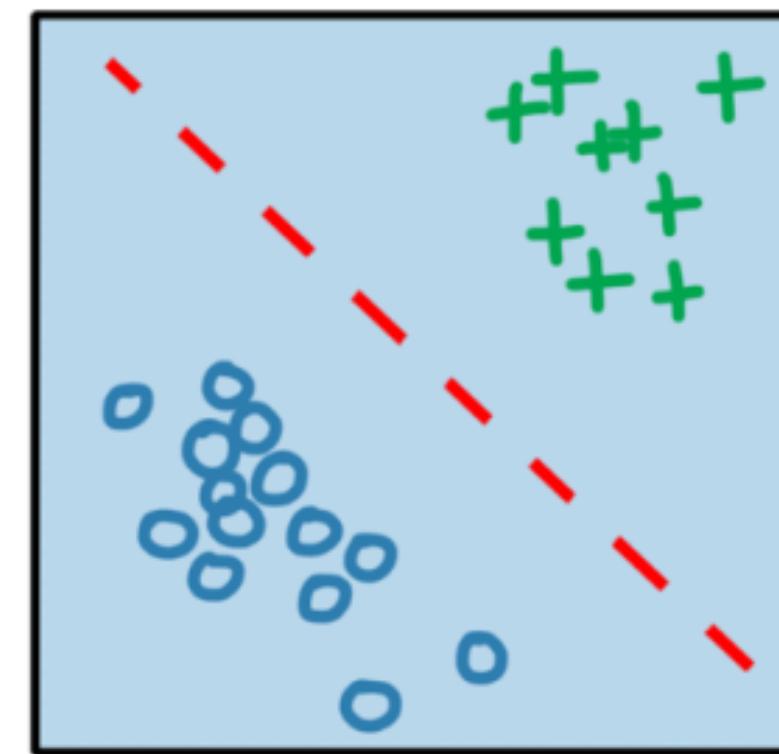


machine learning

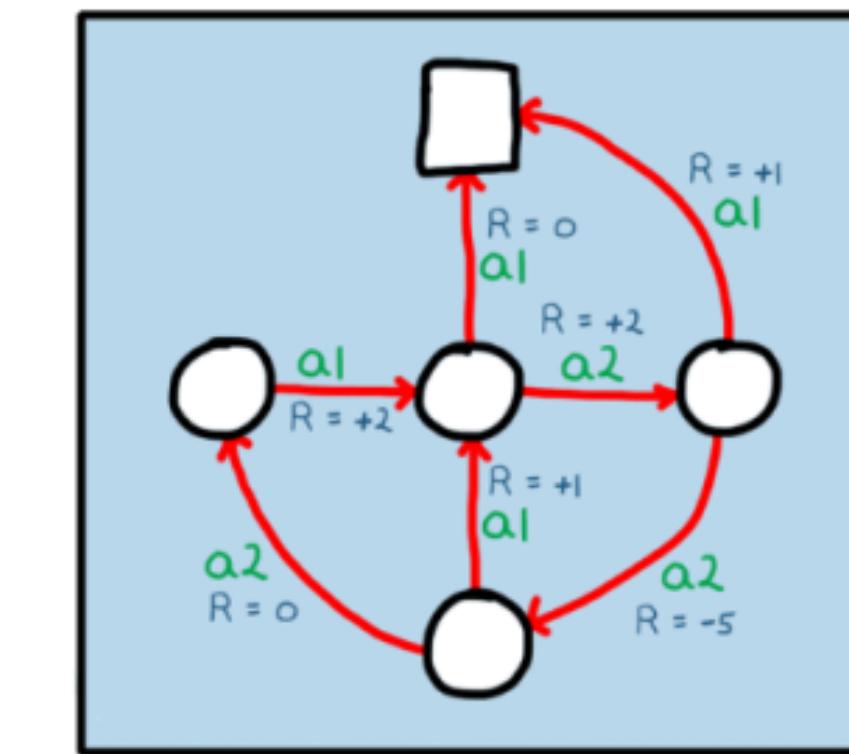
unsupervised
learning

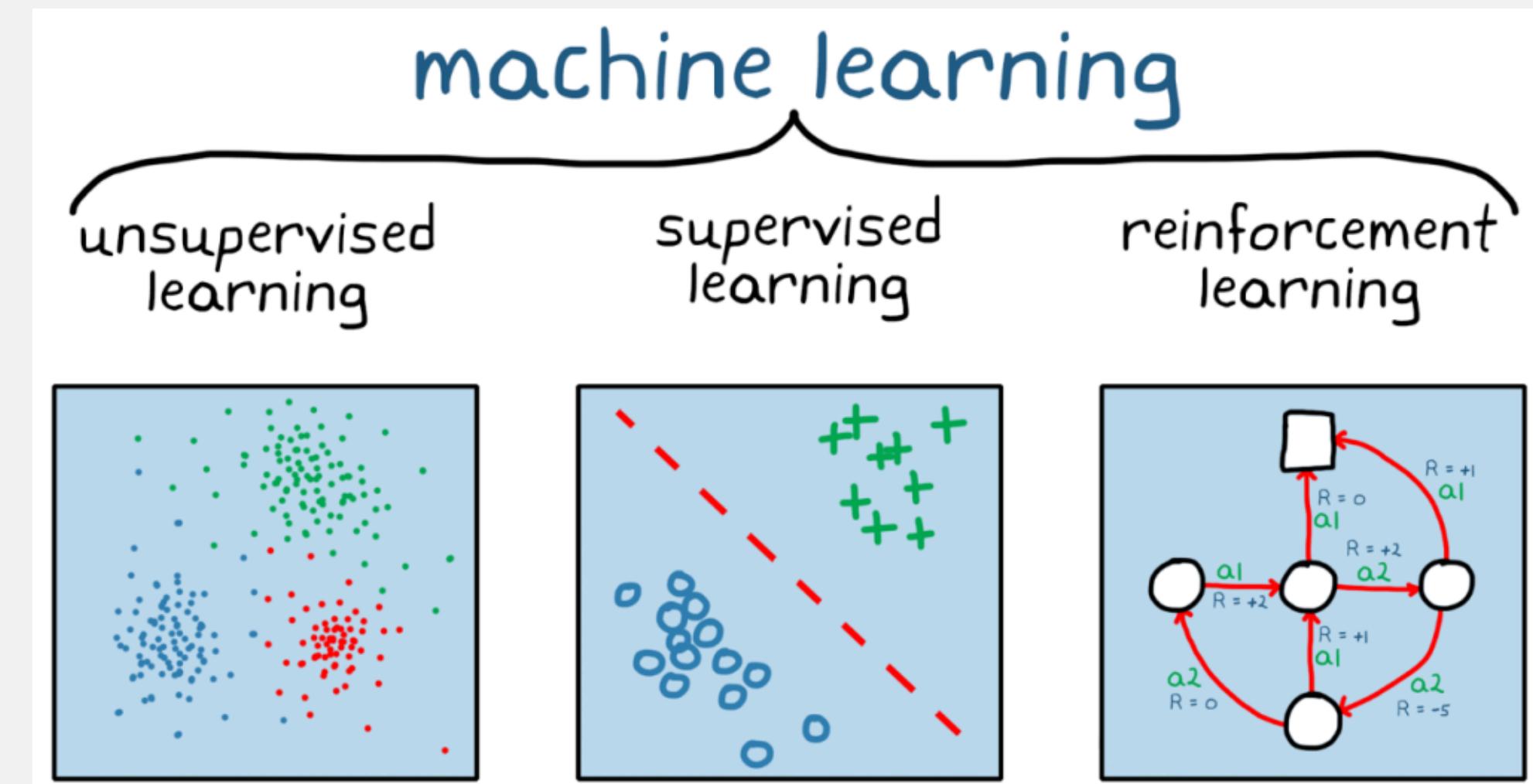


supervised
learning



reinforcement
learning





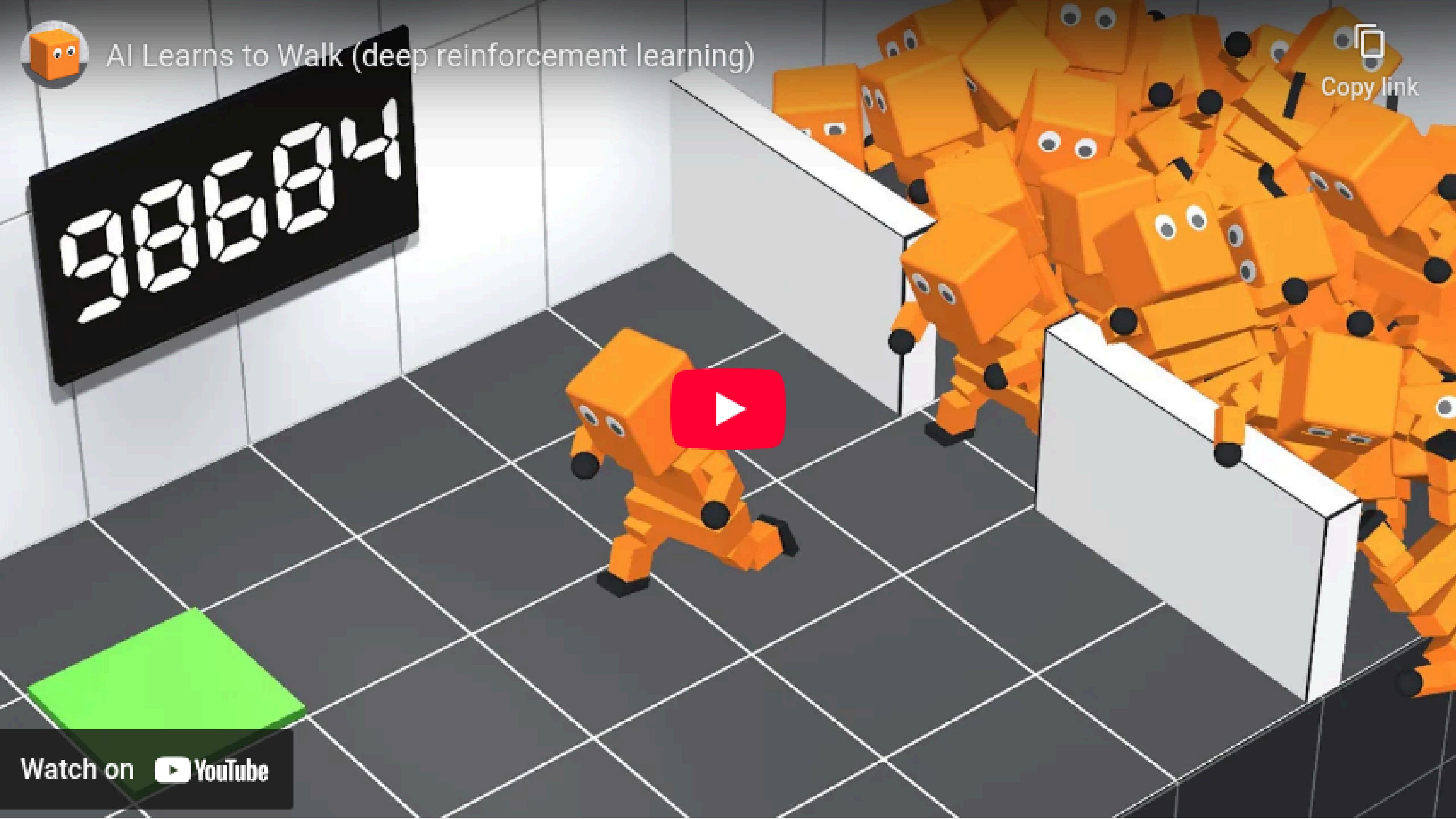
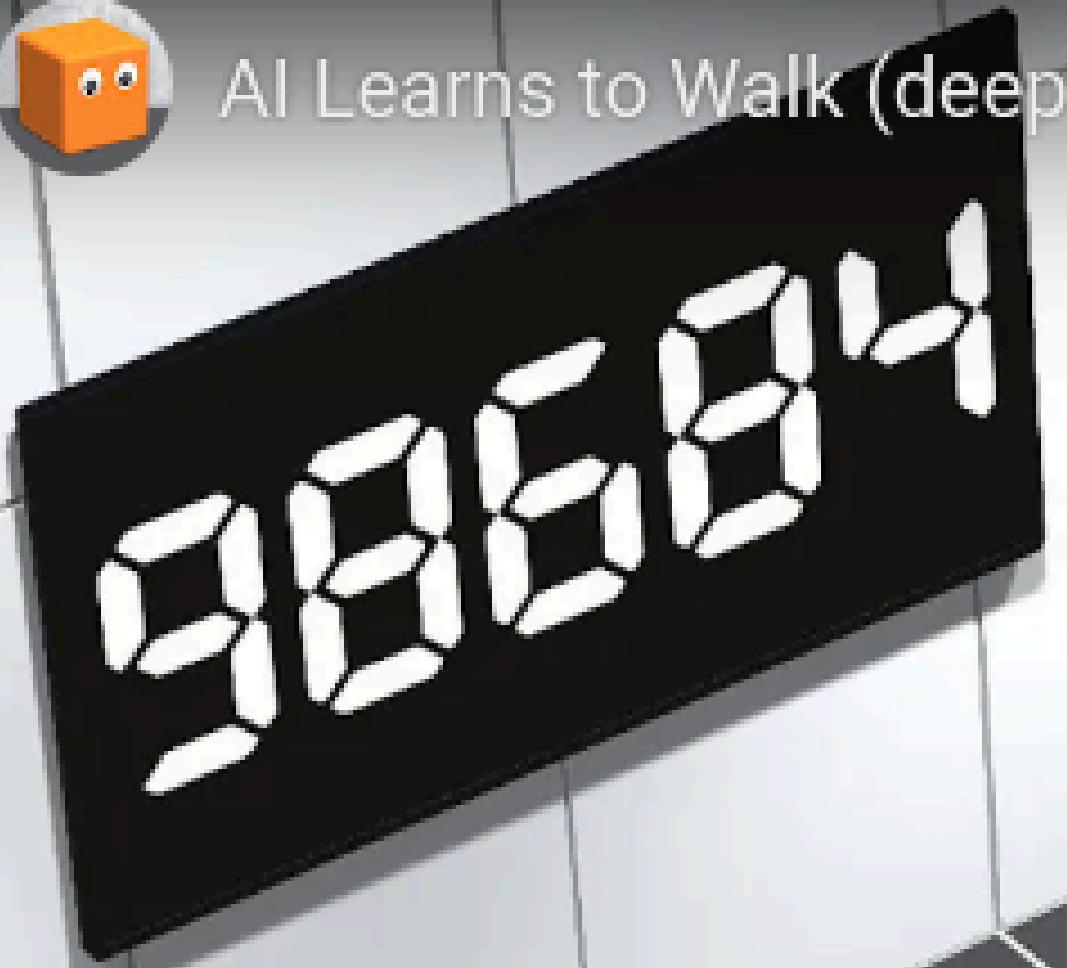
Definition	Finding patterns in data without labeled examples	Learning with labeled examples (we know the correct answers)	Learning through trial and error with rewards/penalties
Examples	<ul style="list-style-type: none"> - Customer segmentation - Market basket analysis - Anomaly detection 	<ul style="list-style-type: none"> - Email spam detection (spam/not spam) - House price prediction - Image recognition 	<ul style="list-style-type: none"> - Game playing (chess, Go) - Robot navigation - Trading strategies



AI Learns to Walk (deep reinforcement learning)



Copy link



Watch on YouTube

INTRO

PIPELINE

TYPES

OTHERS



Let's Code!

YouTube Trending Video Analysis

Goal: Walk through the full pipeline of a mini data science project using real-world YouTube trending data.

INTRO

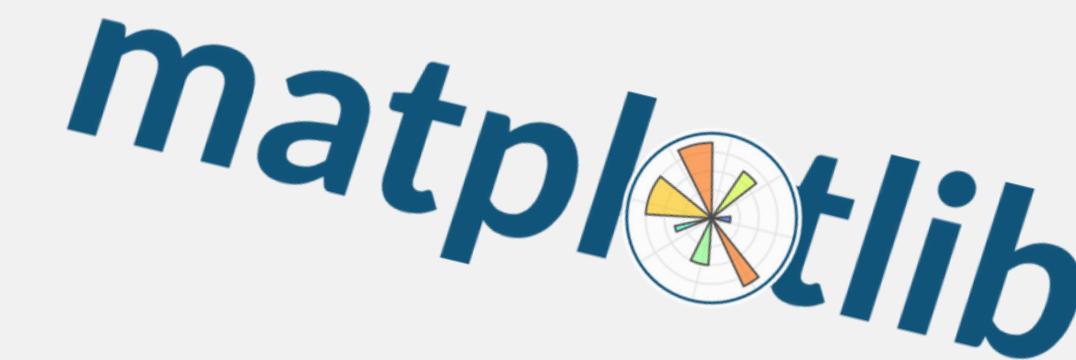
PIPELINE

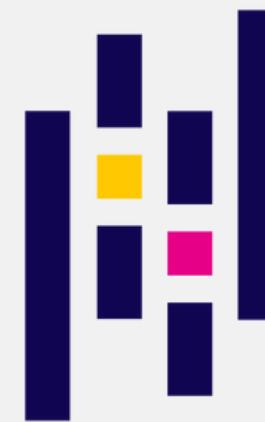
TYPES

OTHERS

BUT! Before that...

Recap for Python Libraries:

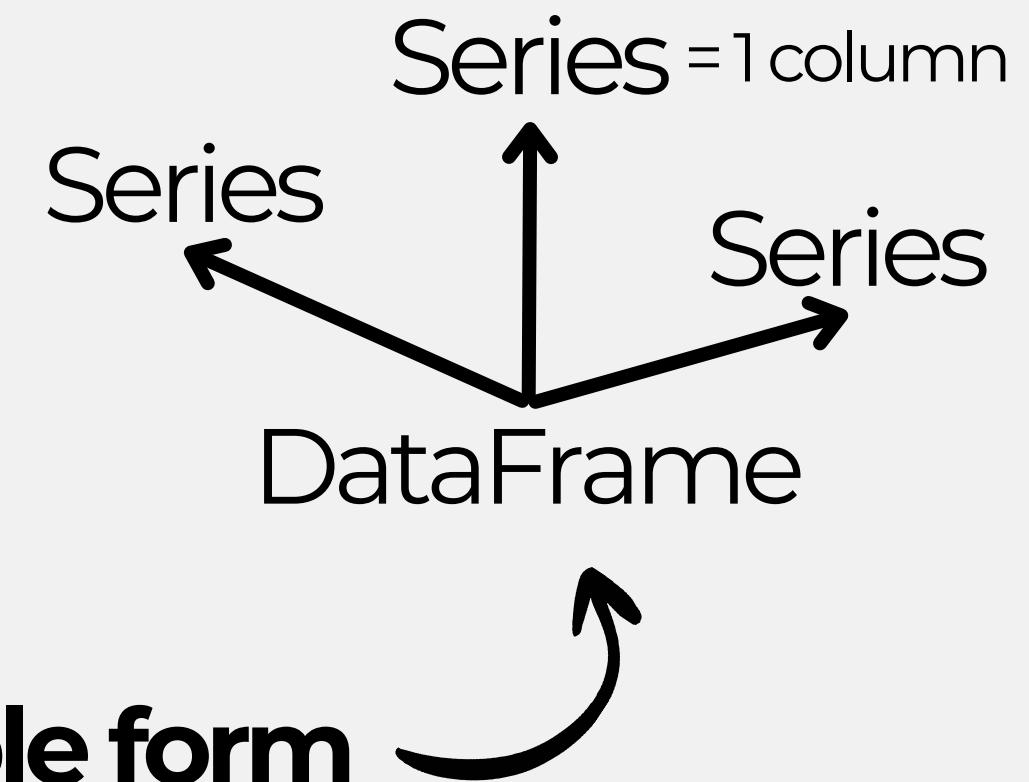




pandas

= python library that **keeps your data in table form**
(like excel/spreadsheet) in memory, so you can do:

- sort
- filter
- group
- join
- etc



INTRO

PIPELINE

TYPES

OTHERS

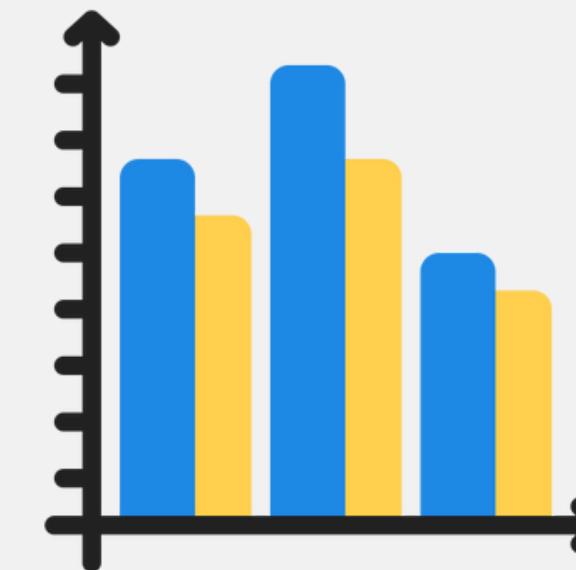
matplotlib



= python library for **data visualization**



data



chart

INTRO

PIPELINE

TYPES

OTHERS

What we'll do later:



pandas



matplotlib

Raw Data

DataFrame

Visualize



Learn SQL in 5 minutes! (Can you?)

learning

SQL

in 5 minutes



?

```
1SELECT COUNT(*) FROM
2(SELECT COUNT(*) FROM
3(SELECT food_type, COUNT(*)
4FROM zoo
5LEFT JOIN food
6 ON food.animal = zoo.animal
7GROUP BY food_type
8HAVING COUNT(*) < 10) AS inner_inner_
```

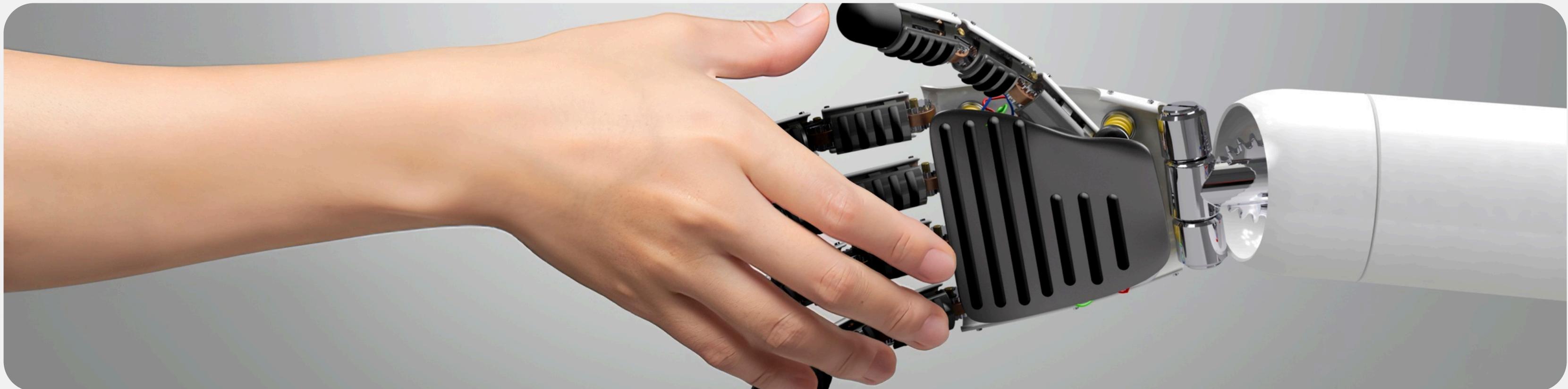
Watch on YouTube

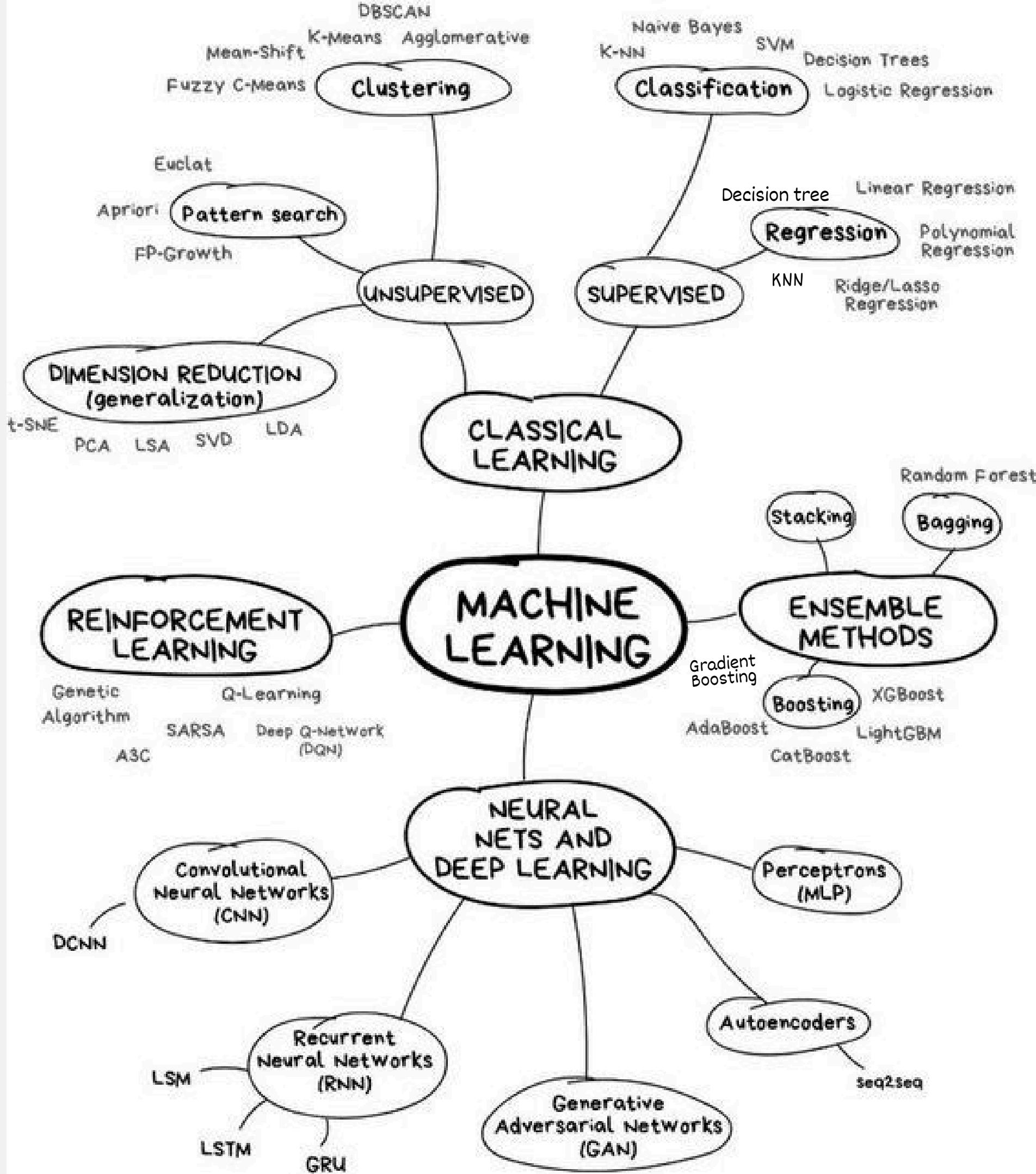


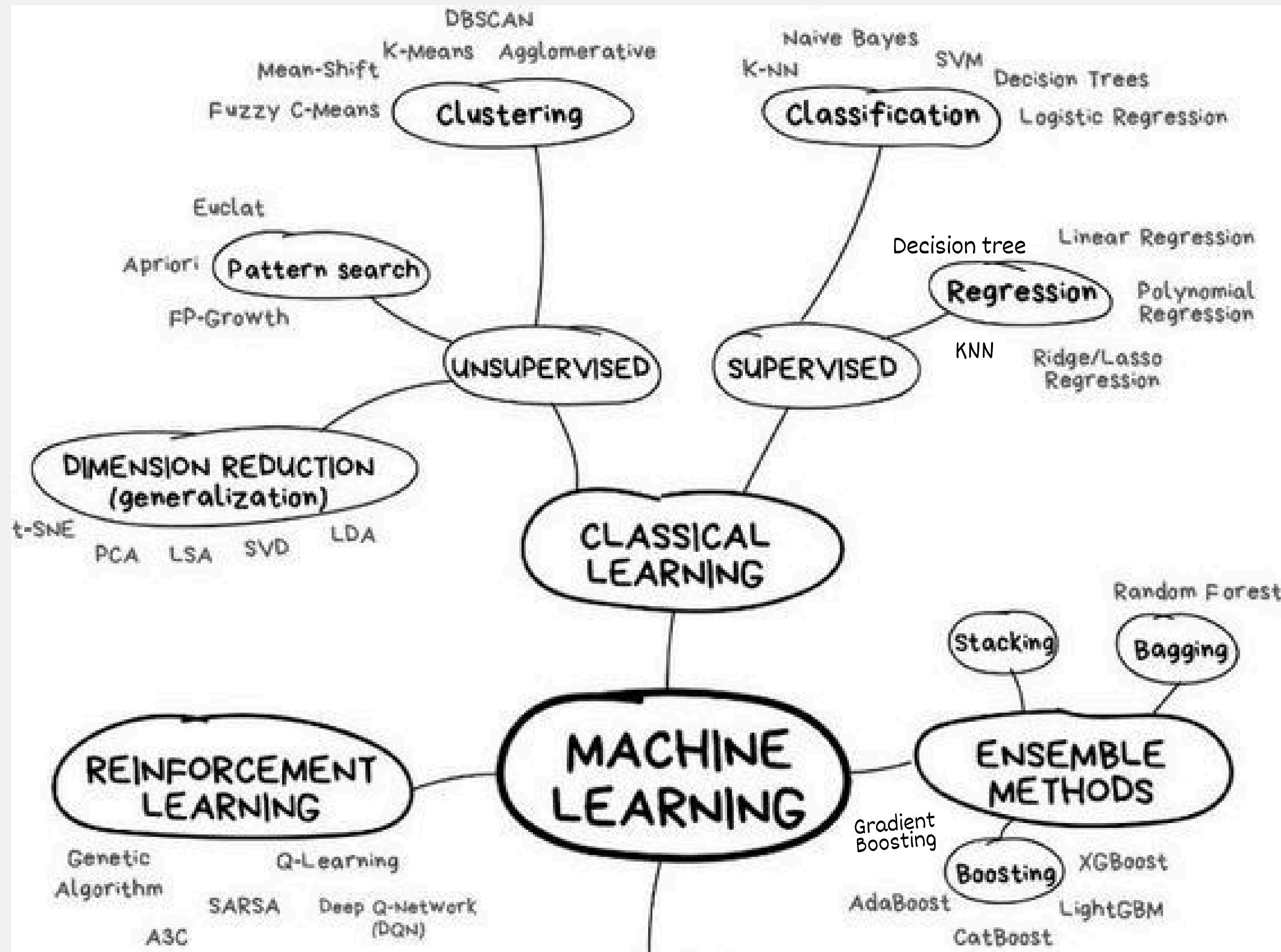
Copy link

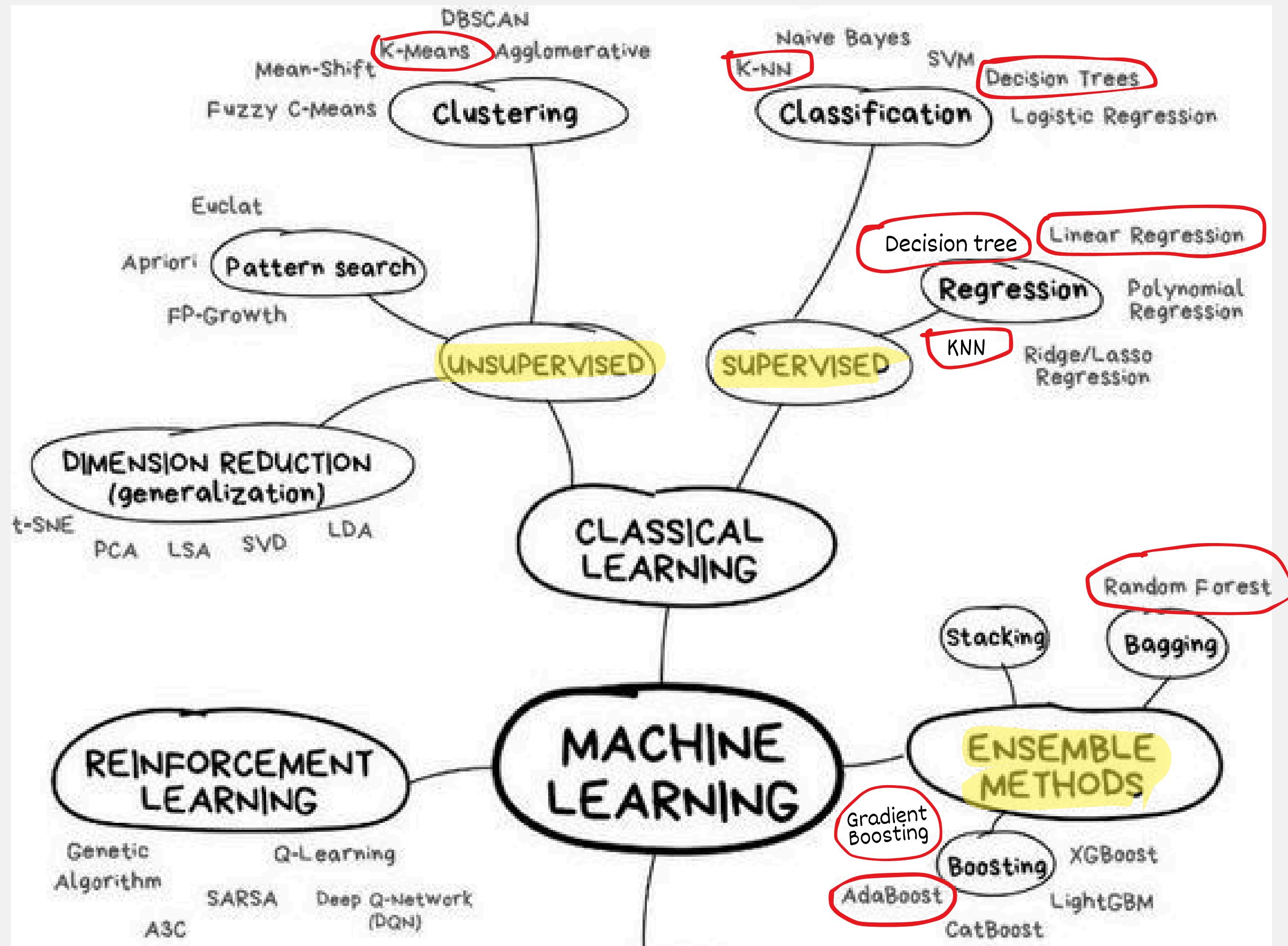
Common Algorithm

= a set of **mathematical rules** and steps that a model
use to learn pattern from data









What is Supervised Learning

(Recap)

- Train with **labelled data**
 - the dataset has the correct answers.
- Learn to predict outcomes (numbers or categories).



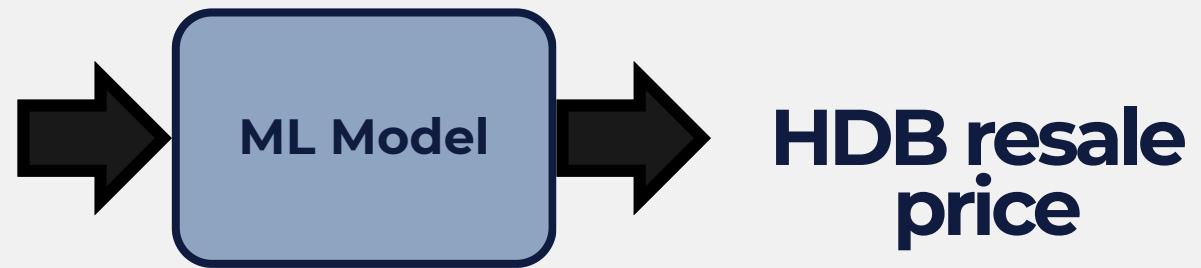
Types of Outcome

(for Supervised Learning)

Regression

(Predict Number)

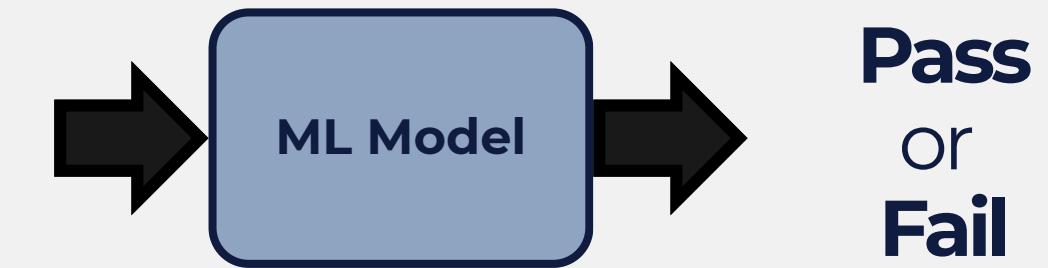
Remaining lease,
Floor area,
Location ,
etc



Classification

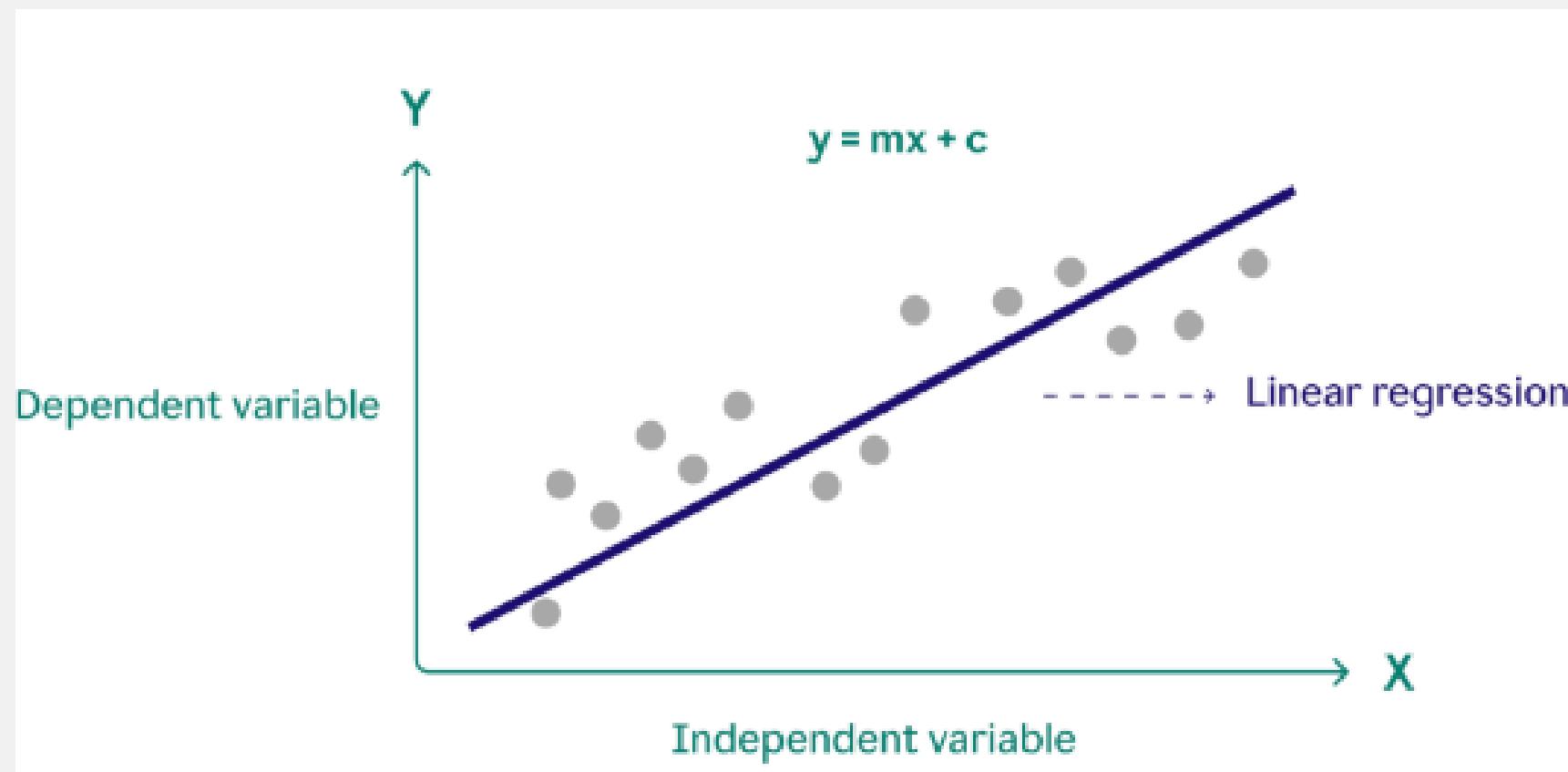
(Predict Category)

Hours studied,
Attendance rate,
Assignment score,
etc



Linear Regression

(for Supervised Learning)



Strengths	Weaknesses
Simple and easy to interpret	Struggles with non-linear patterns
Works well when the relationship is linear	Sensitive to outliers

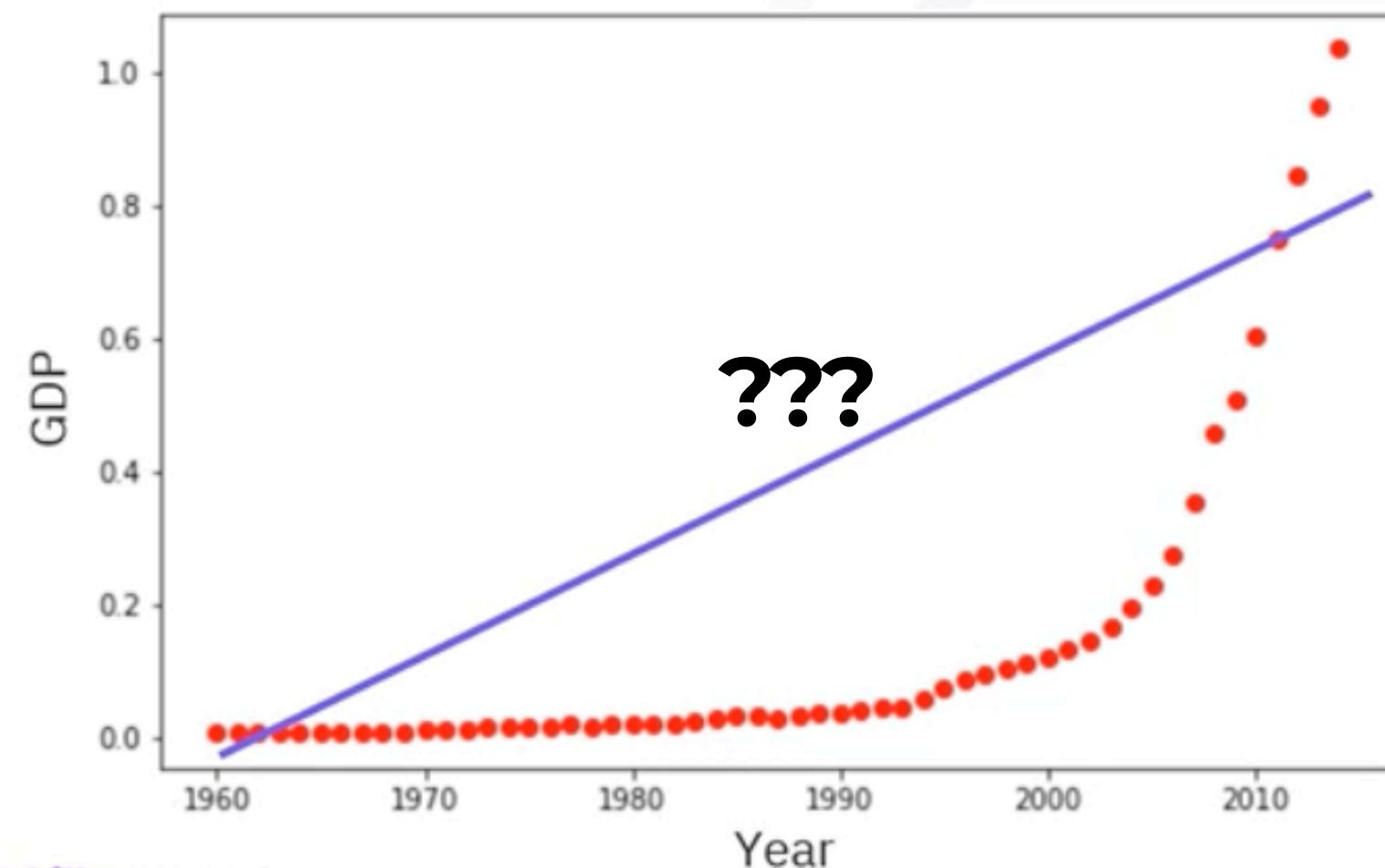
Definition: A model that predicts a continuous value based on the relationship between input features and the target.

Use Case: Predicting prices, trends (e.g., house price prediction).

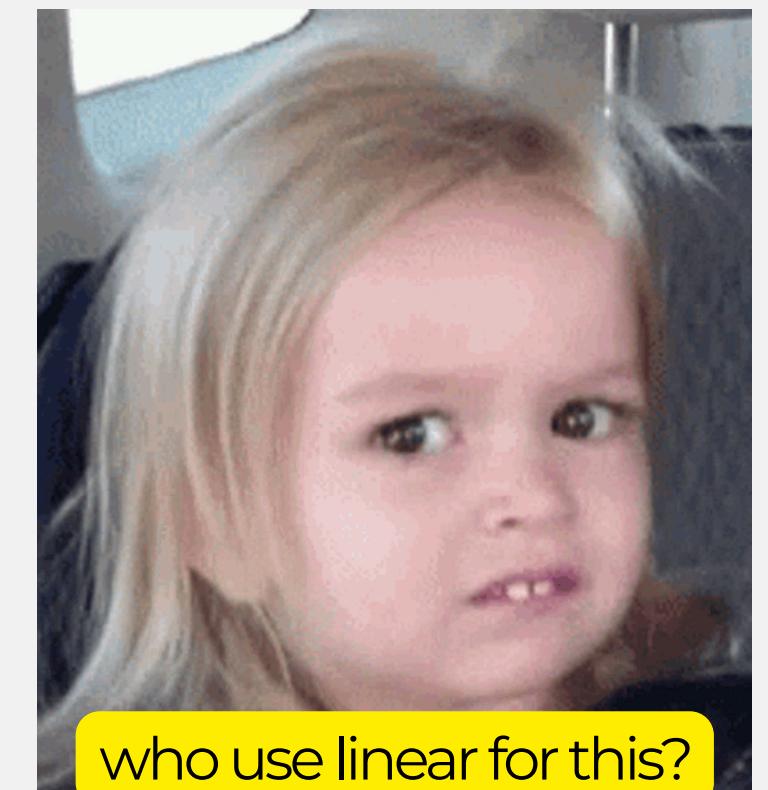
IRL, its not as simple as linear..

Compound growth example

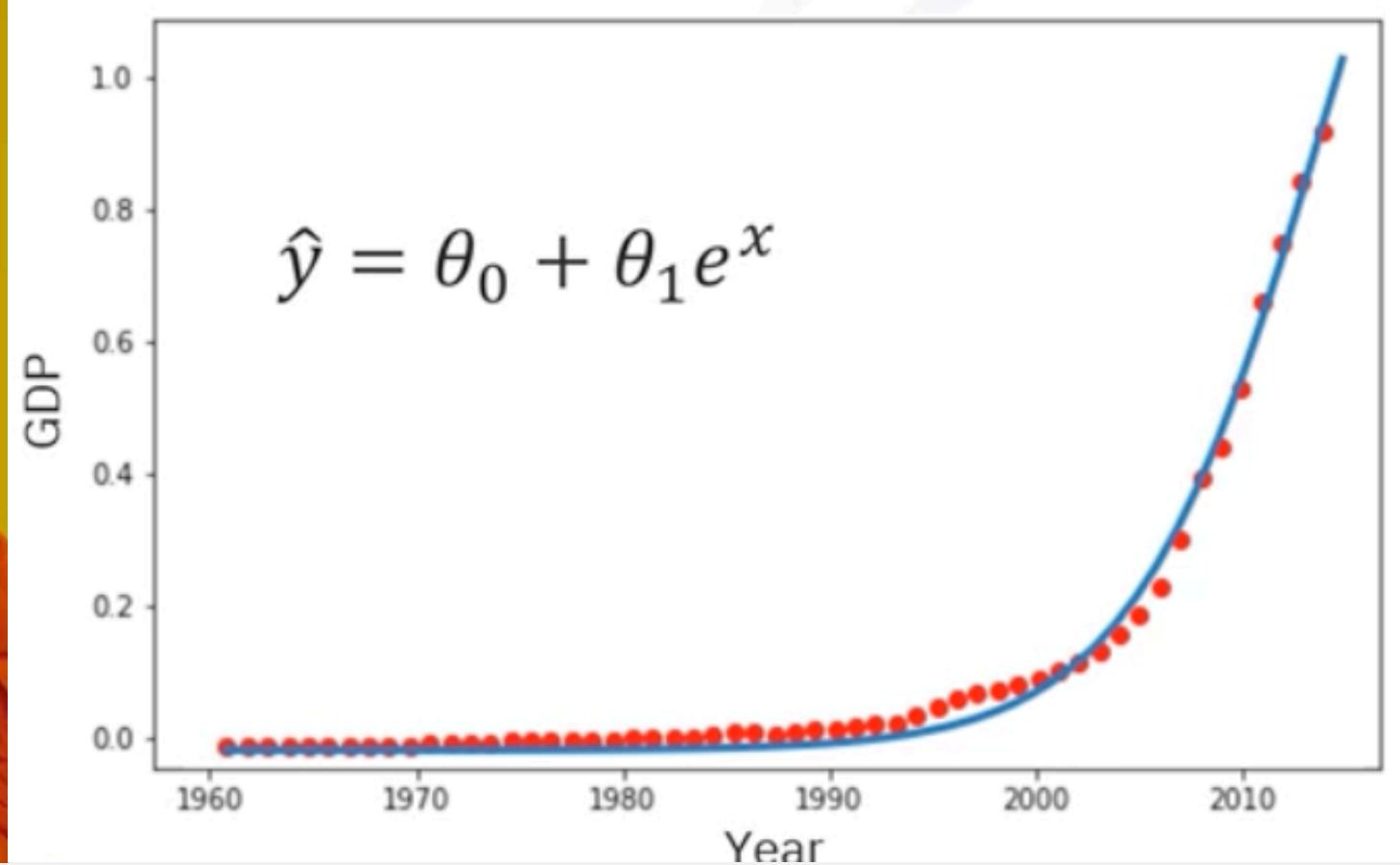
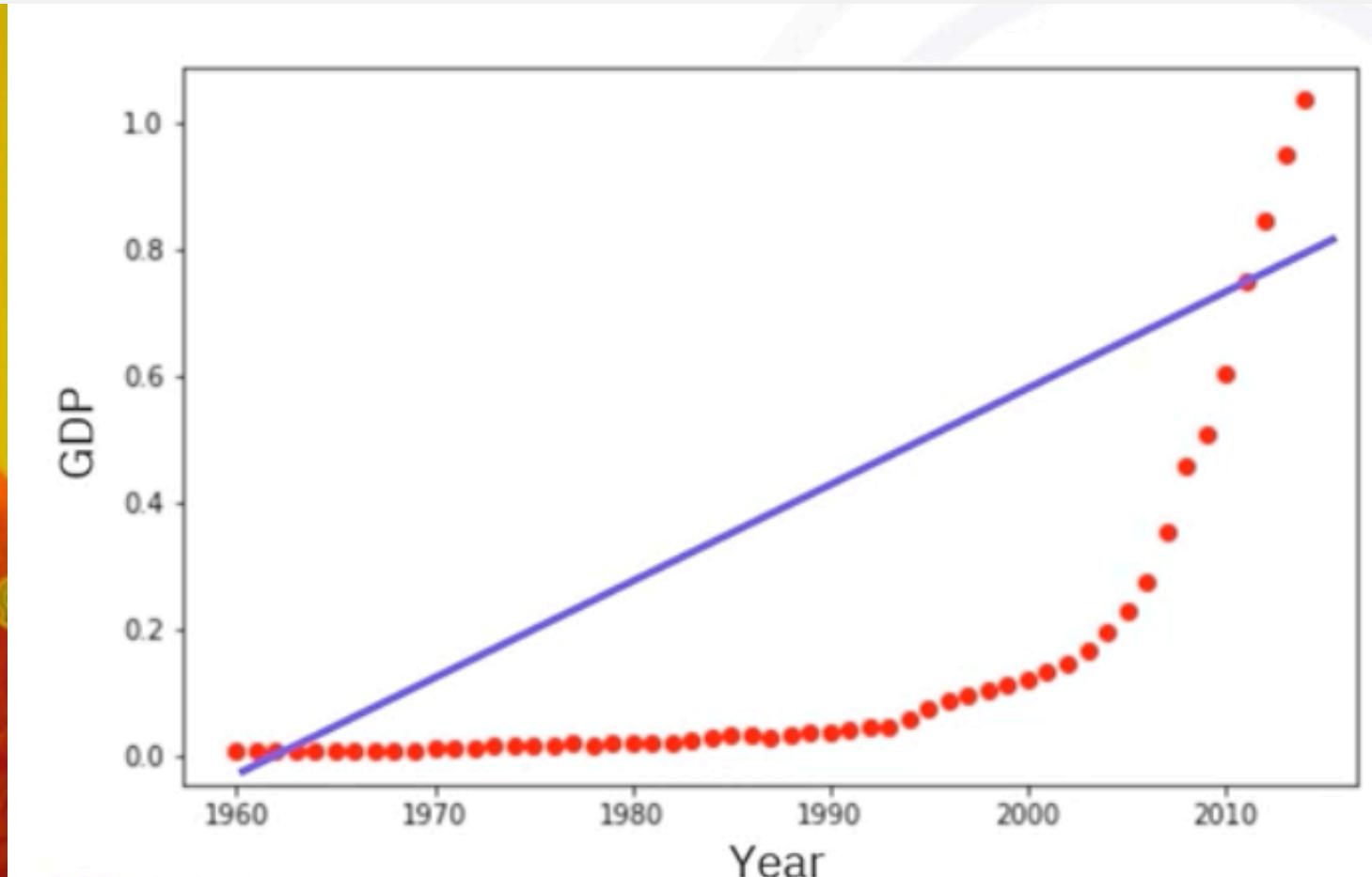
- Scatterplot displays strong dependence of GDP on time, but relationship is nonlinear



	Year	GDP (B USD)
0	1960	59.2
1	1961	49.6
2	1962	46.7
3	1963	50.1
4	1964	59.1
5	1965	69.8
6	1966	75.9
7	1967	72.1
...



who use linear for this?



Non-Linear Regression

(an extension of Linear Regression)

Polynomial Regression

- Quadratic – 2nd order

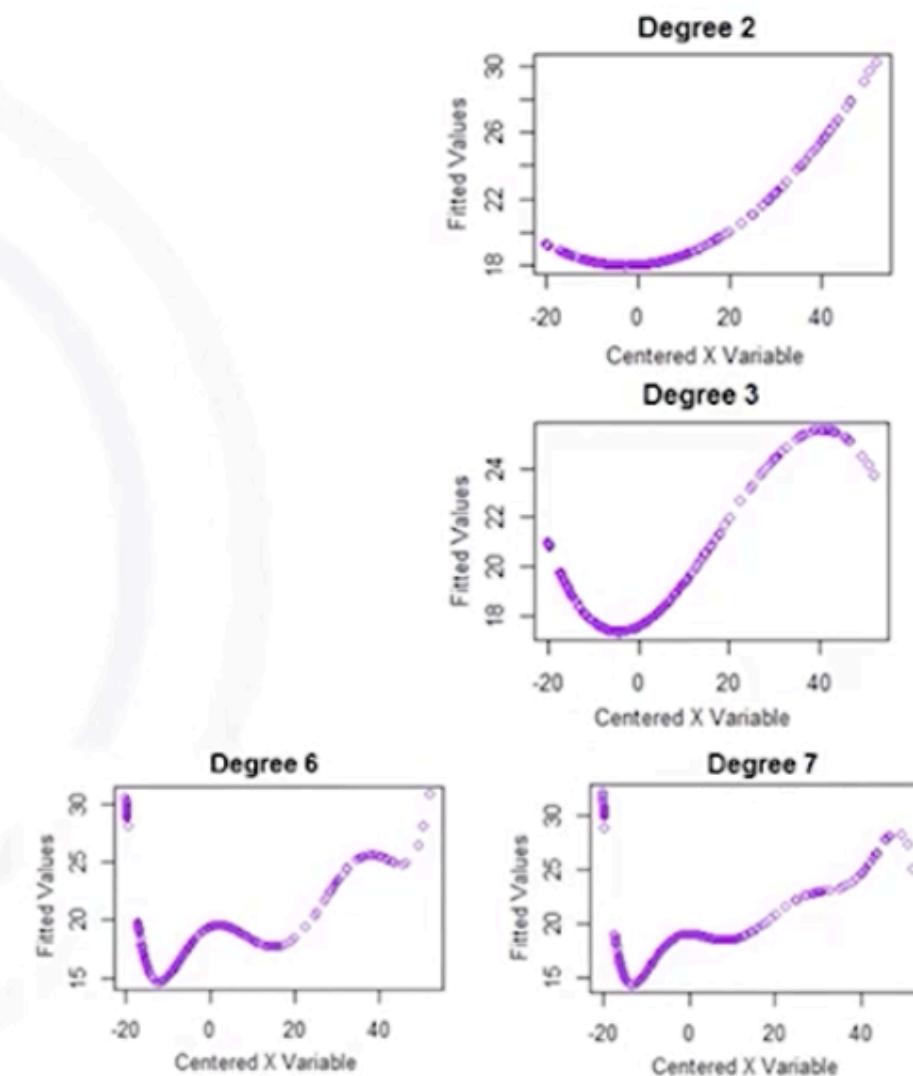
$$\hat{Y} = b_0 + b_1 x_1 + b_2 (x_1)^2$$

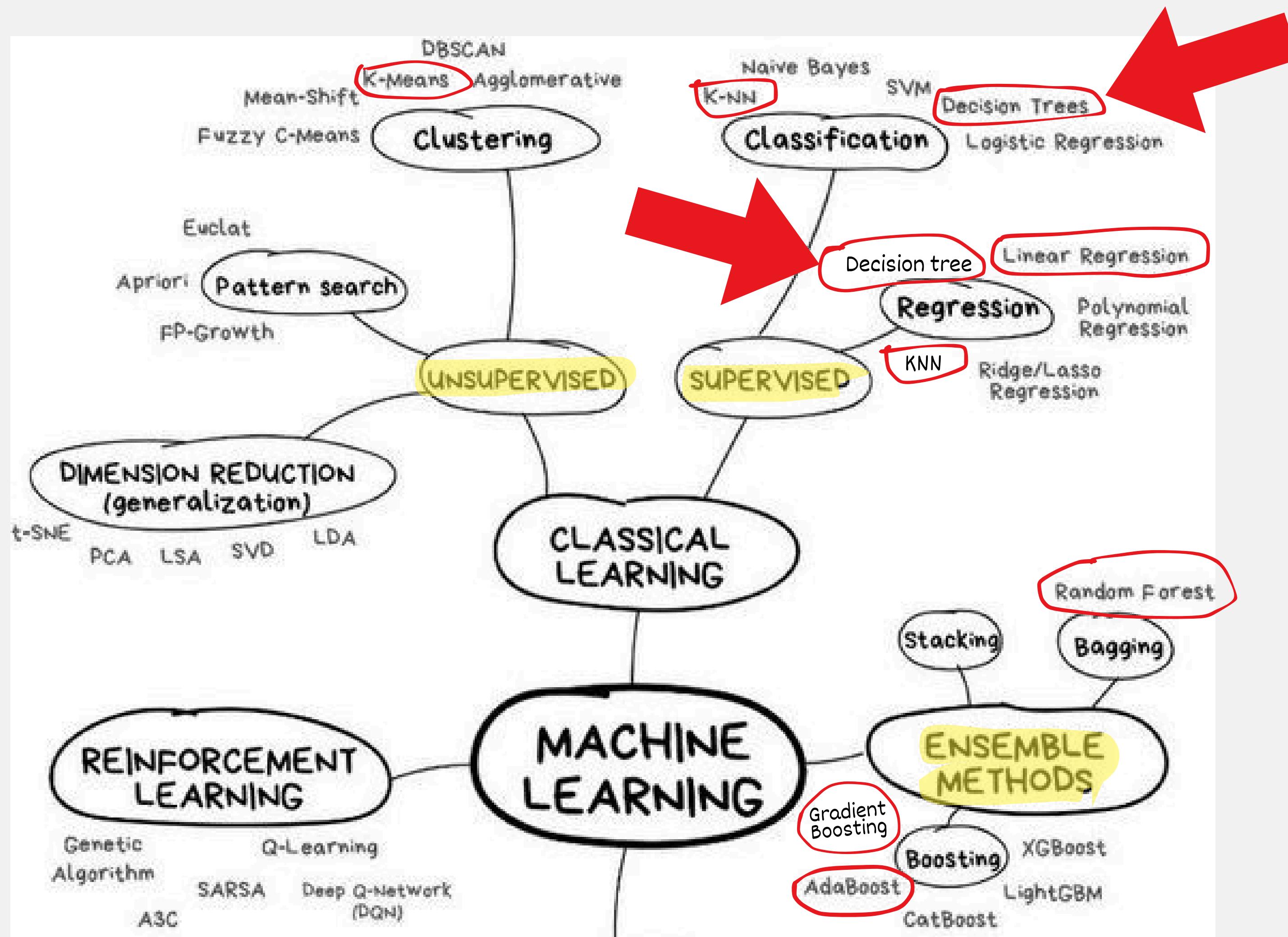
- Cubic – 3rd order

$$\hat{Y} = b_0 + b_1 x_1 + b_2 (x_1)^2 + b_3 (x_1)^3$$

- Higher order

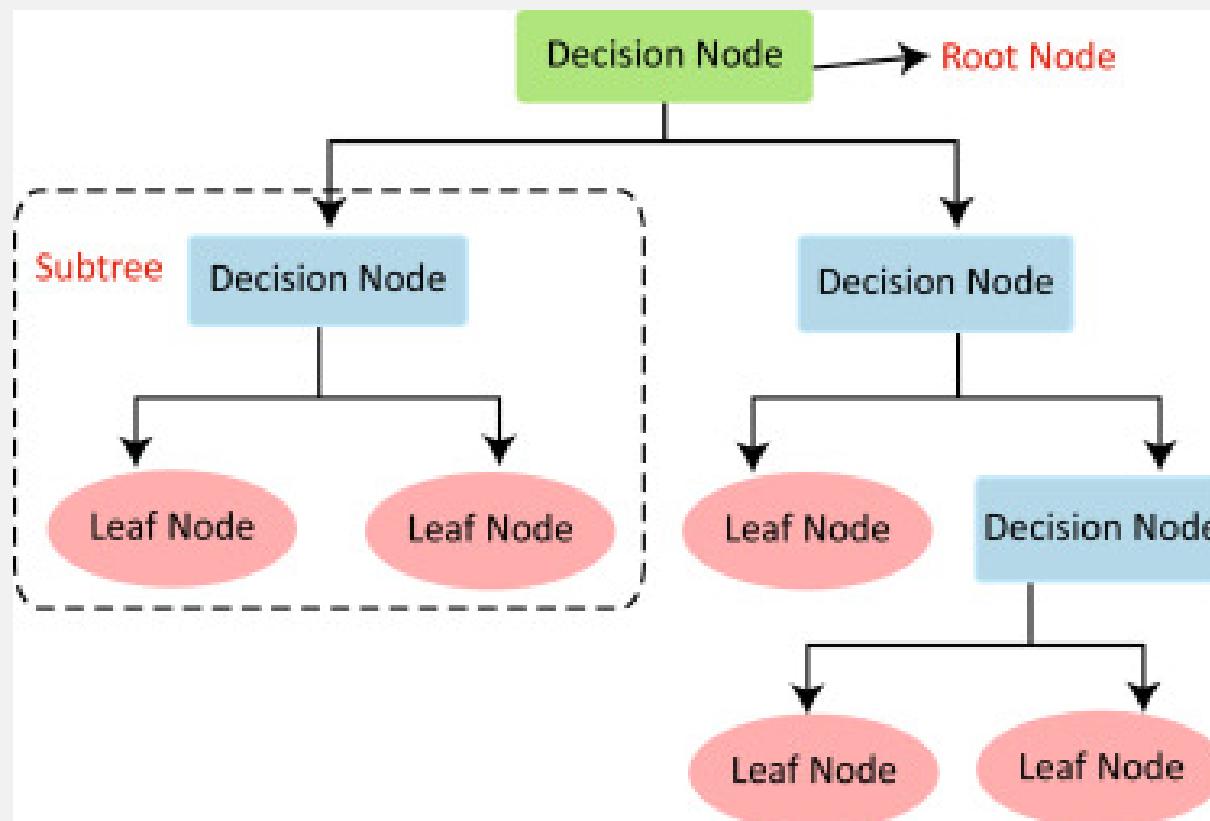
$$\hat{Y} = b_0 + b_1 x_1 + b_2 (x_1)^2 + b_3 (x_1)^3 + \dots$$





Decision tree

(For regression and classification)



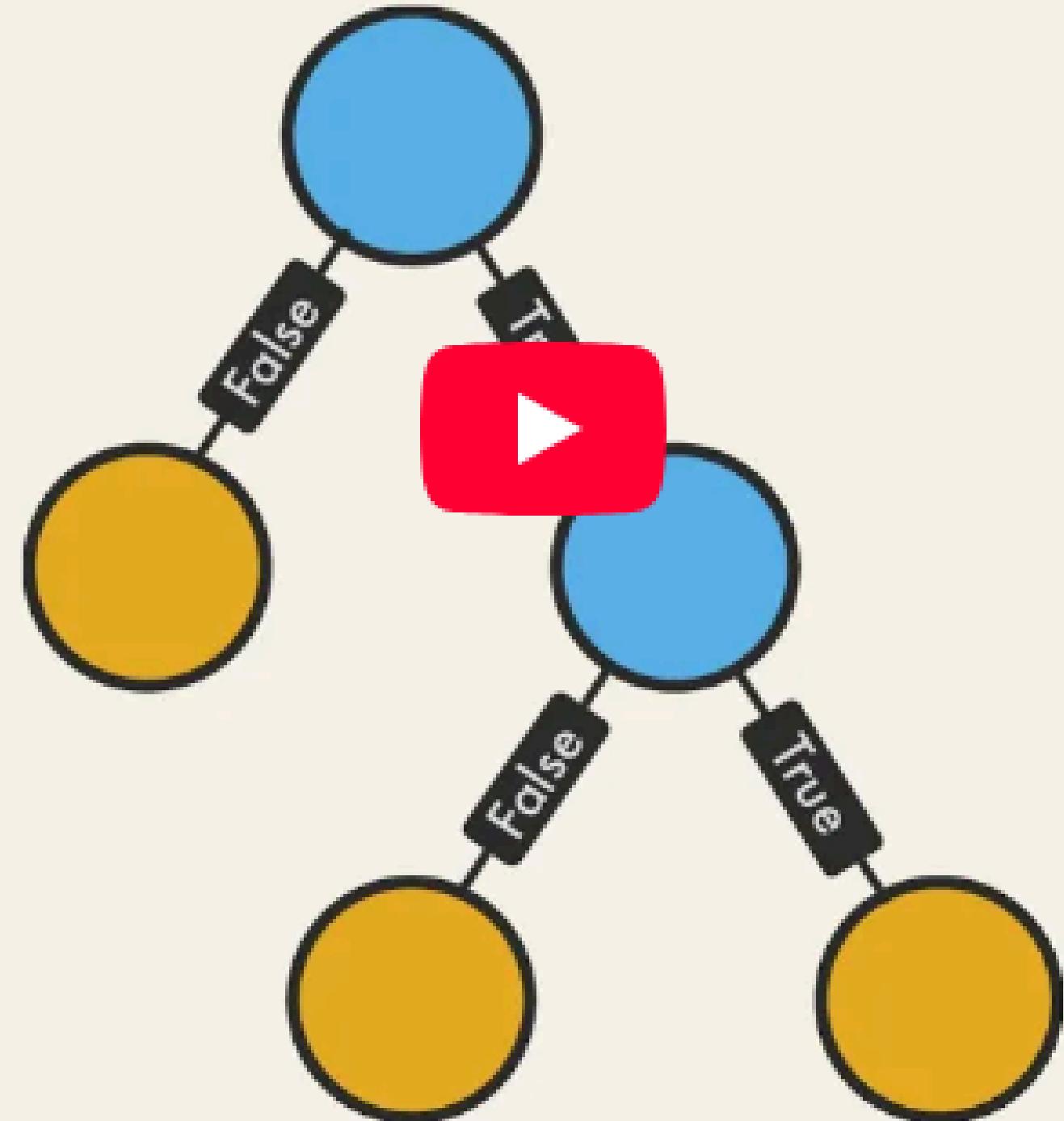
Strengths	Weaknesses
Easy to understand and visualize	Prone to overfitting
Can handle both numerical and categorical data	Small changes in data can lead to a completely different tree

Definition: A tree-based model that splits the data based on feature values to make decisions.

Use Case: Classification problems like determining whether an email is spam or not.



Decision Tree

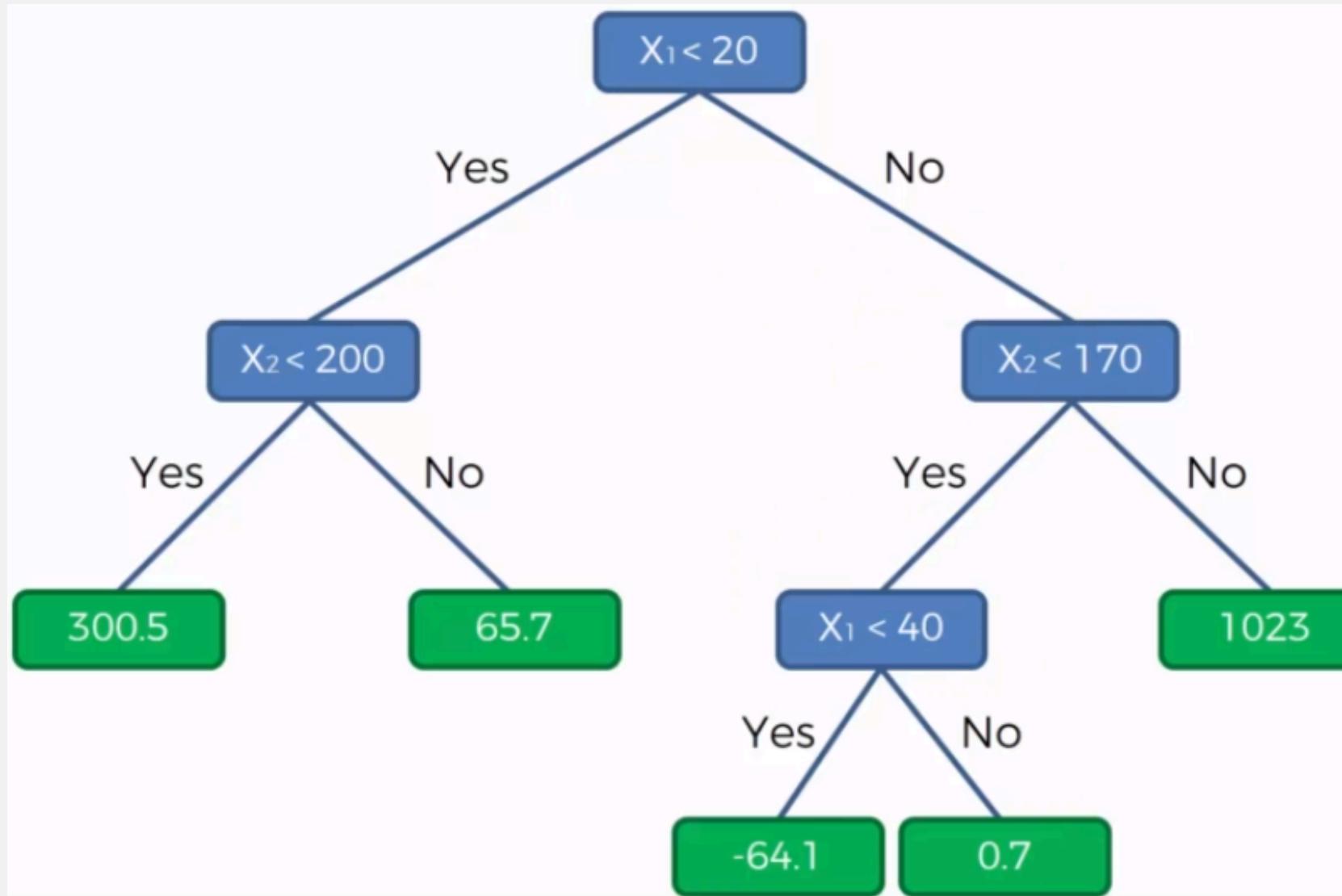


Types of Decision tree

(for Supervised Learning)

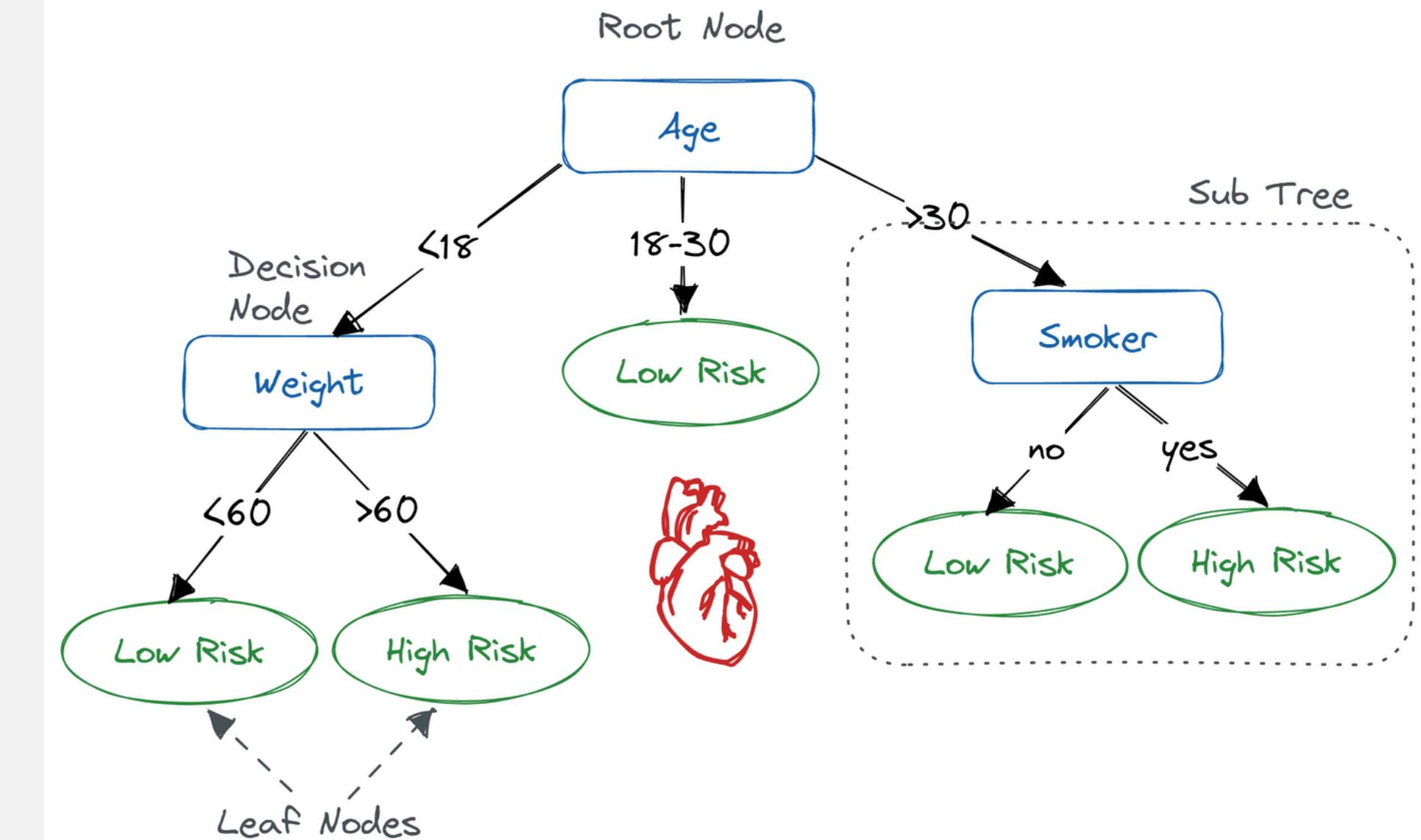
Regressor

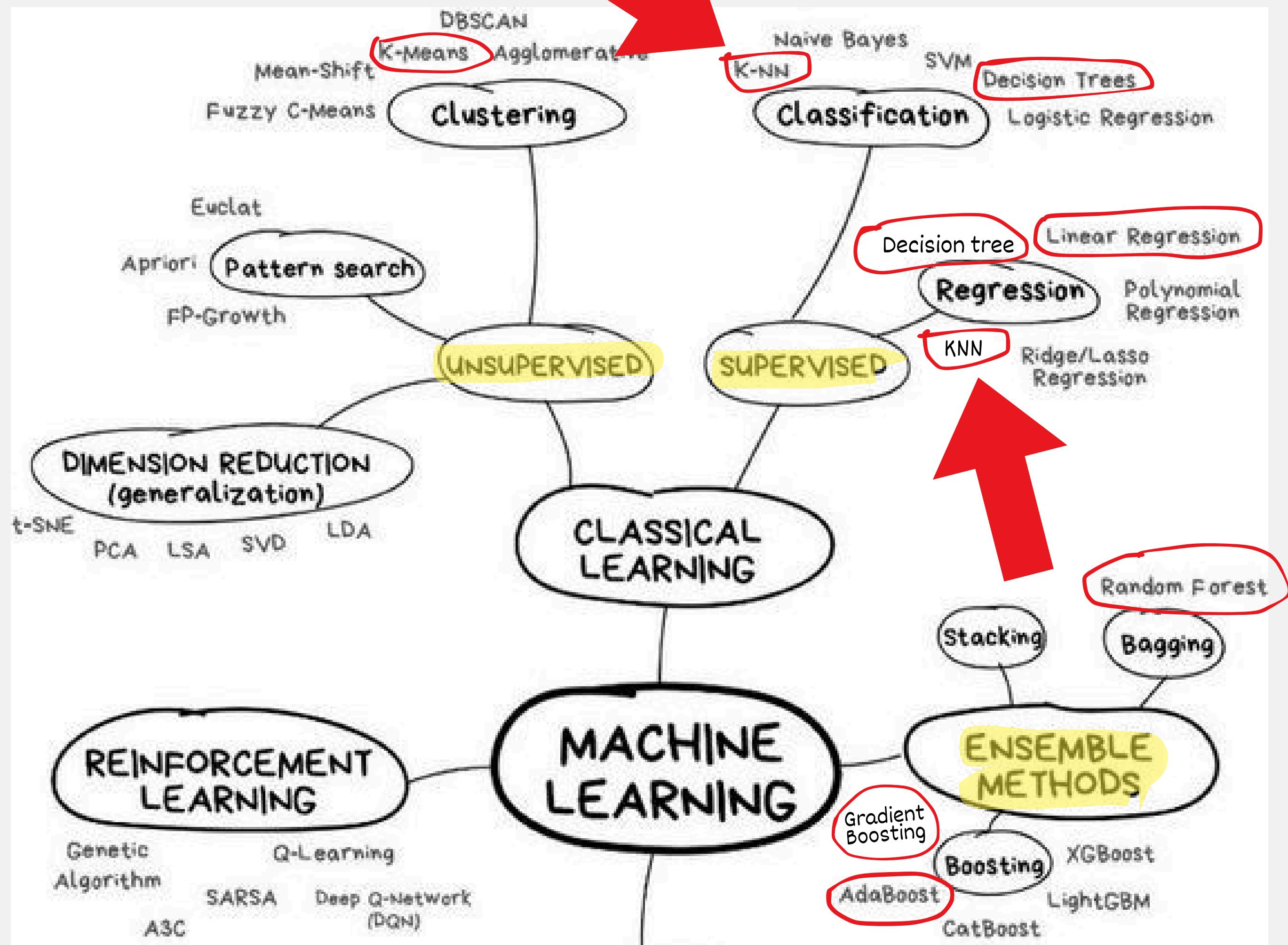
(Predict Number)



Classifier

(Predict Category)





K-Nearest Neighbours (K-NN)

(for Supervised Learning)

- It predicts based on the labels of its nearest neighbors in the training data.

Strengths	Weaknesses
No training phase (just stores data)	Sensitive to irrelevant features or different scales
Works for both classification and regression.	Slow for large datasets (must compare to all points)

Use Case: “Users who liked this movie also liked...”

- KNN finds users with similar viewing histories 

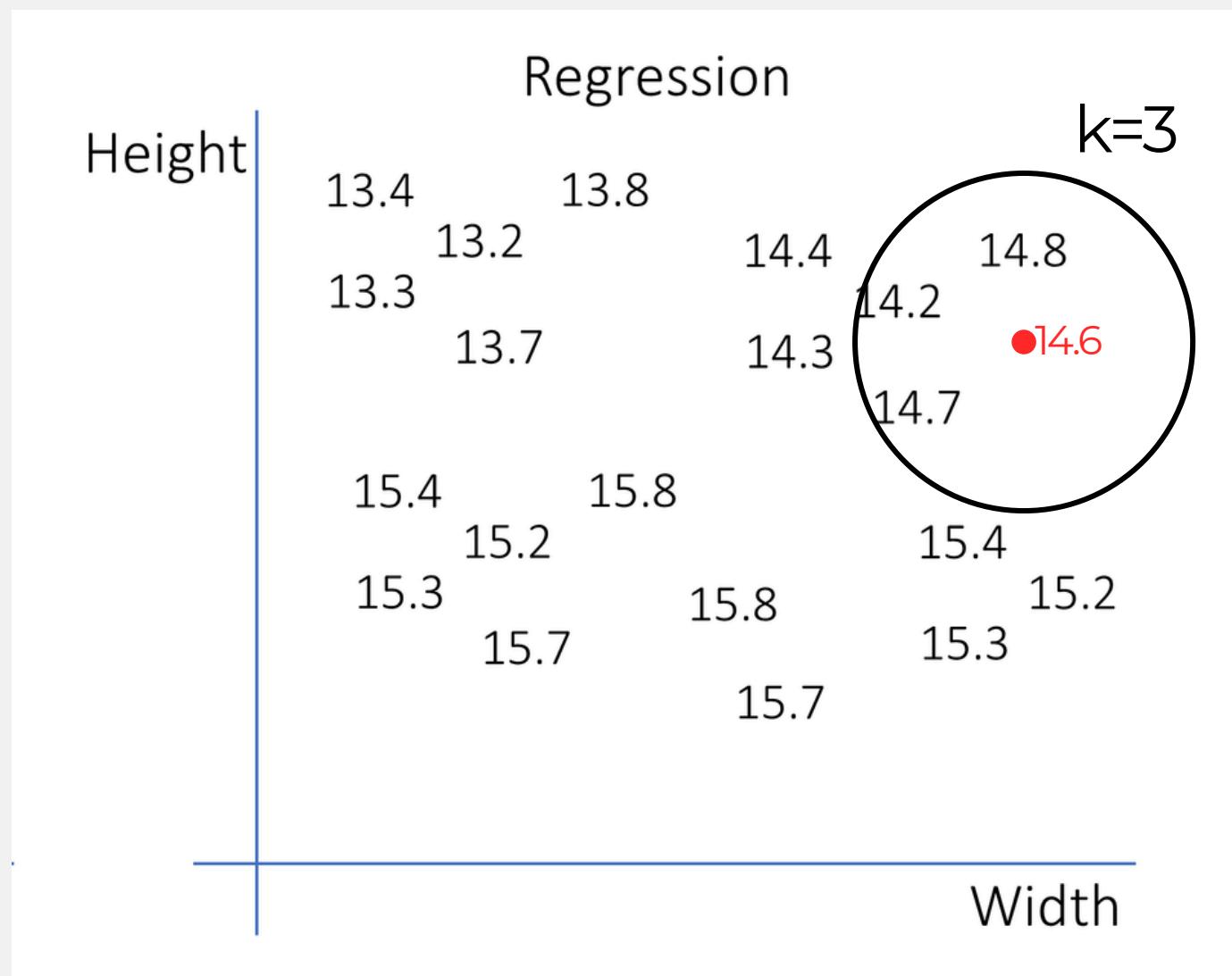


Types of KNN

(for Supervised Learning)

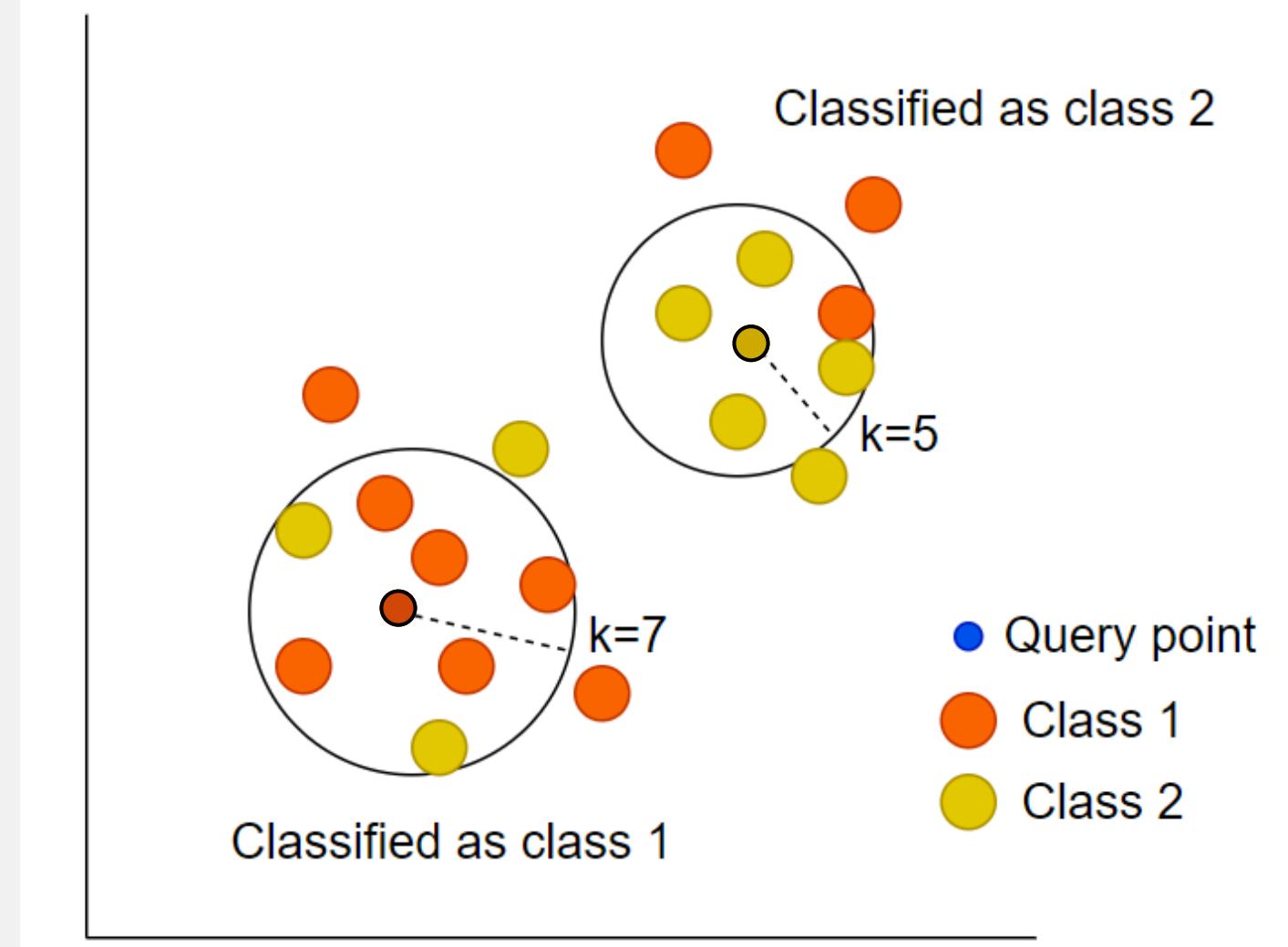
Regressor

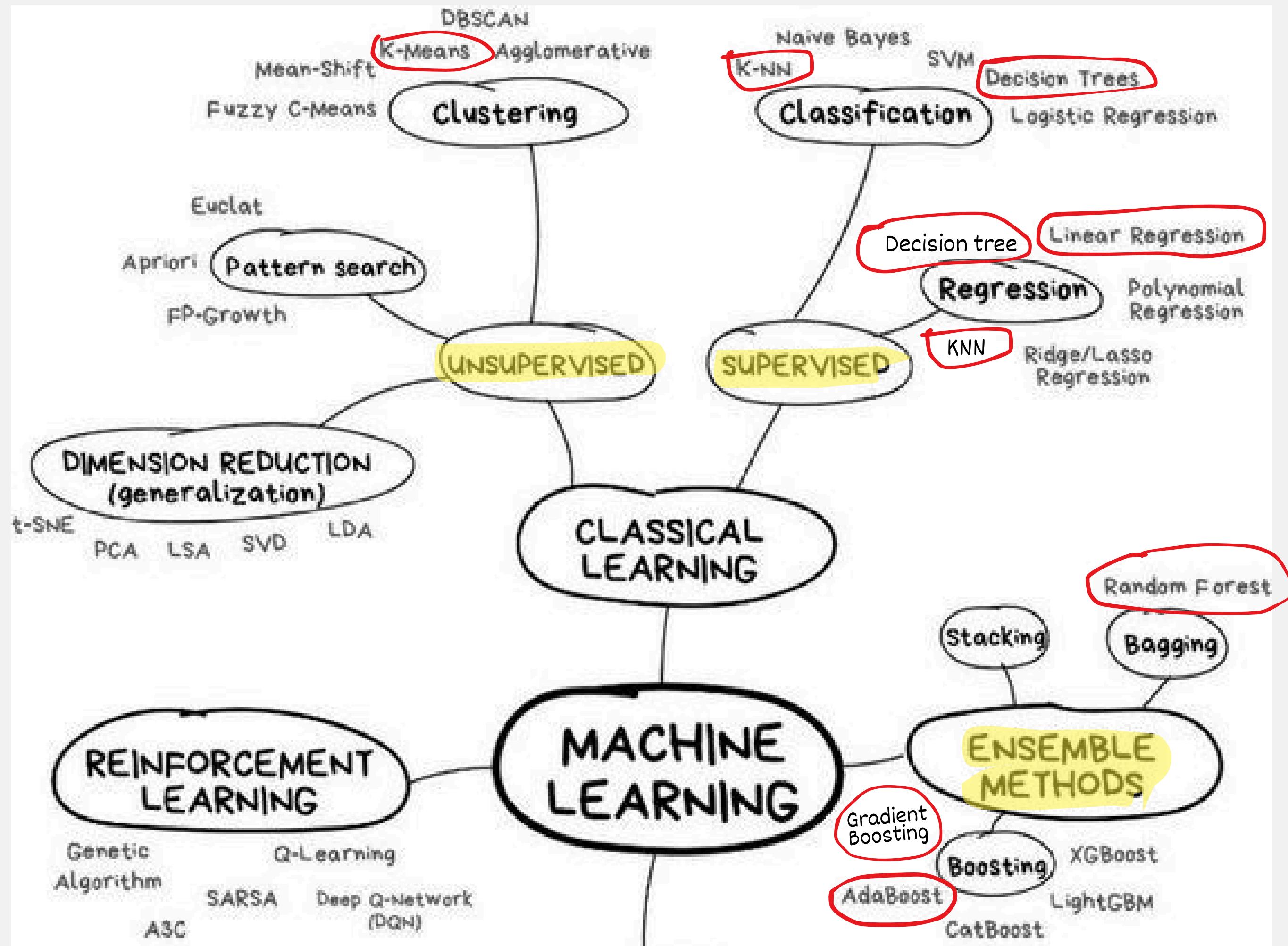
average (or median) of neighbors' values



Classifier

by majority vote of neighbors' labels

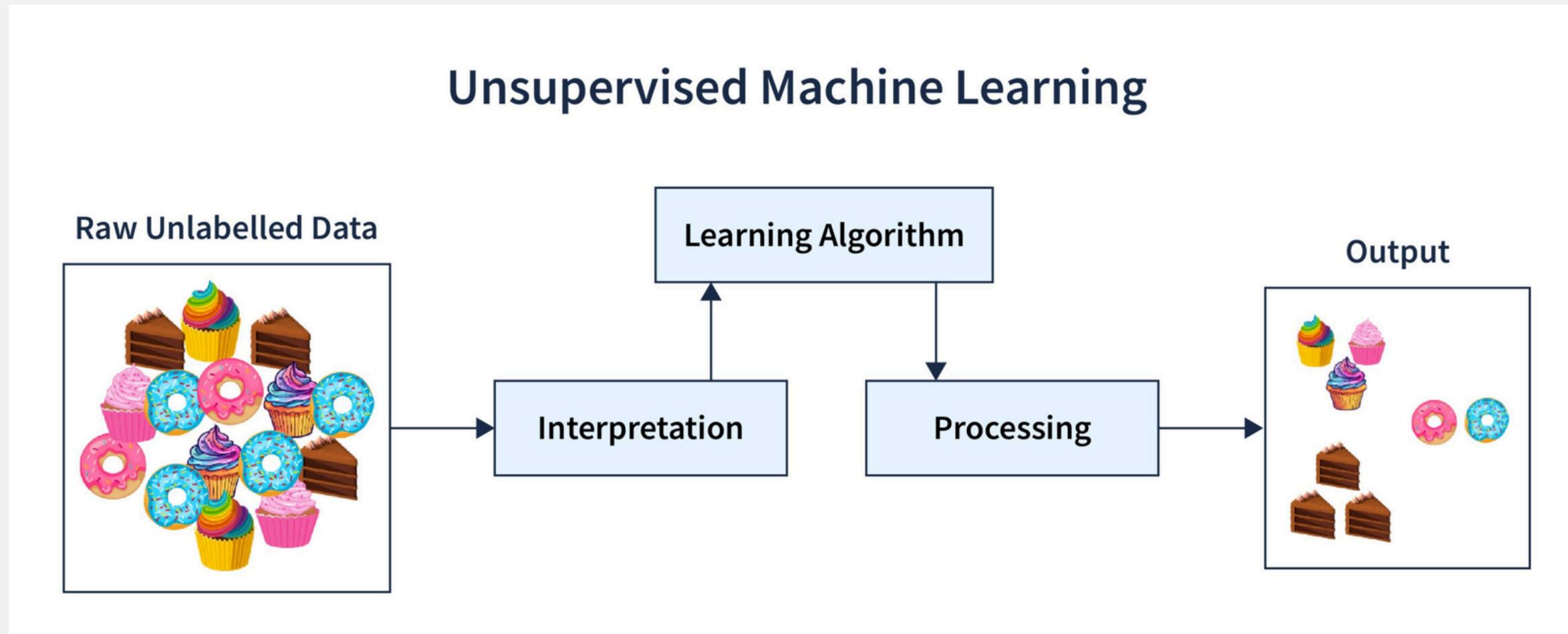


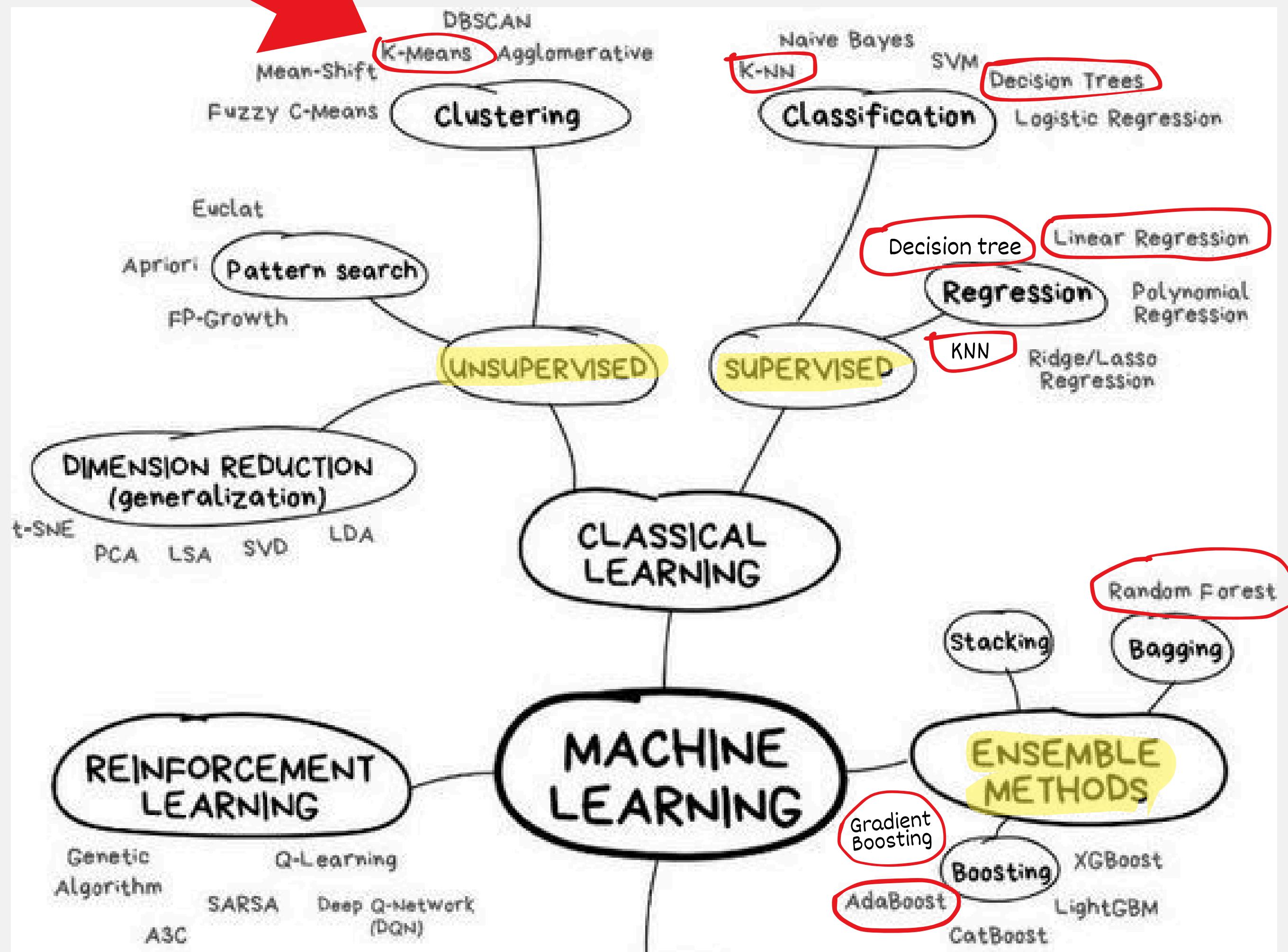


What is Unsupervised Learning

(Recap)

- You train with unlabelled data
 - There are no given answers.
- Goal: find hidden patterns or group similar data.







Classification

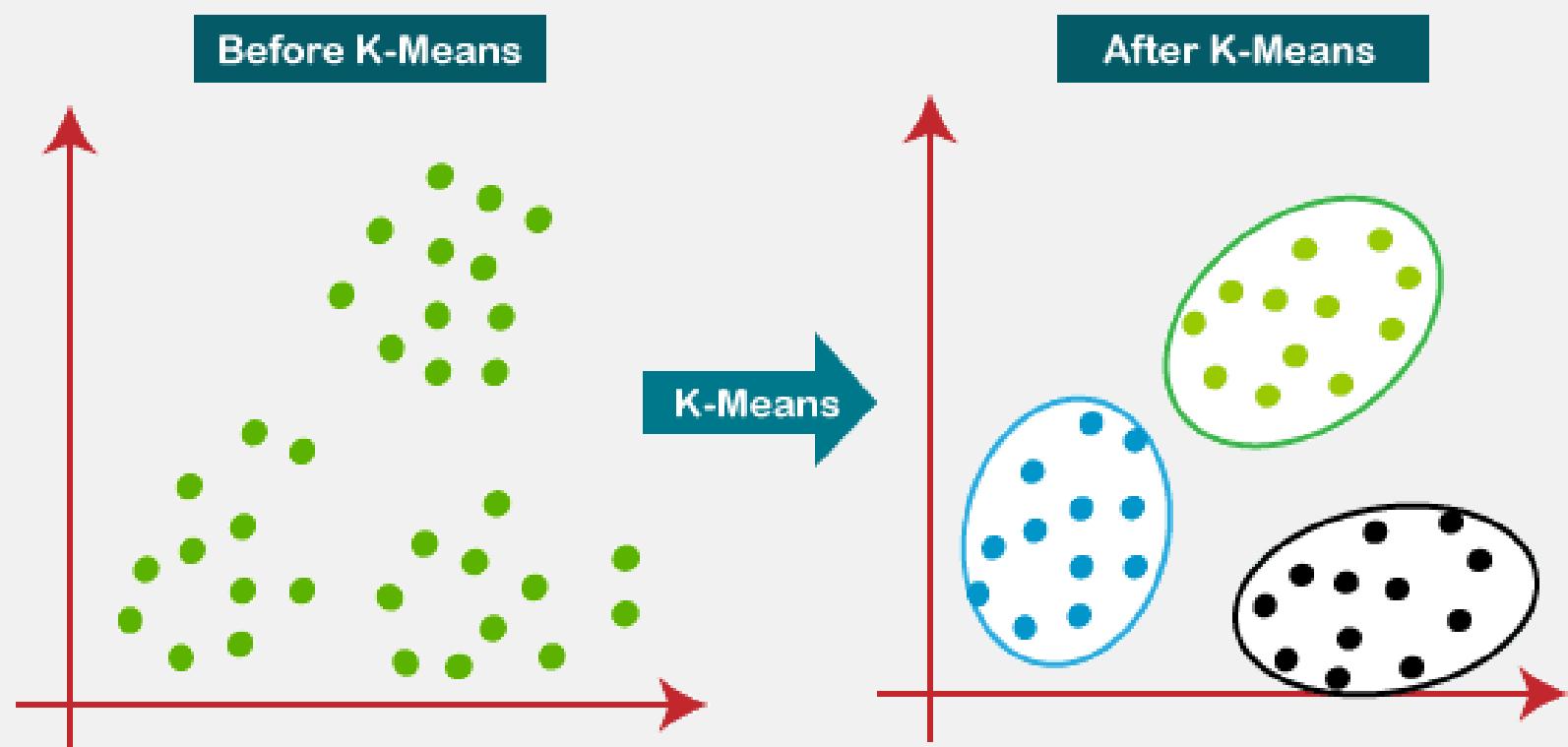
- You already know what categories exist
 - You train your model to recognize them.

Clustering

- You don't know the categories
 - You just want to group similar items.

K-Means Clustering

(for Unsupervised learning)



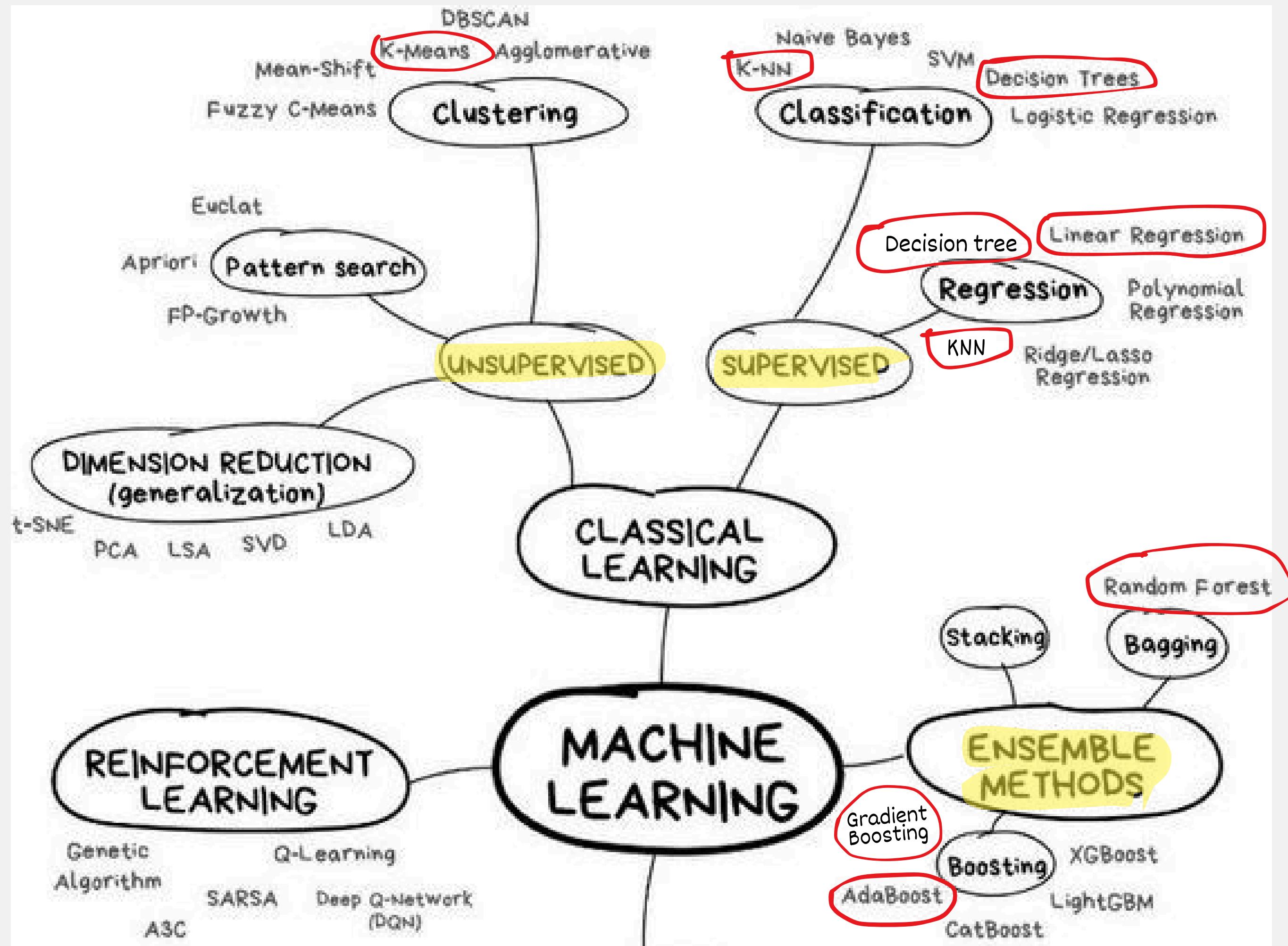
Strengths	Weaknesses
Fast and efficient for large datasets	Requires choosing K manually
Easy to understand and implement	Doesn't work well with non-spherical clusters or different cluster sizes

Definition: An unsupervised algorithm that groups data into K clusters based on similarity.

Use Case: Customer segmentation, grouping similar users or products.

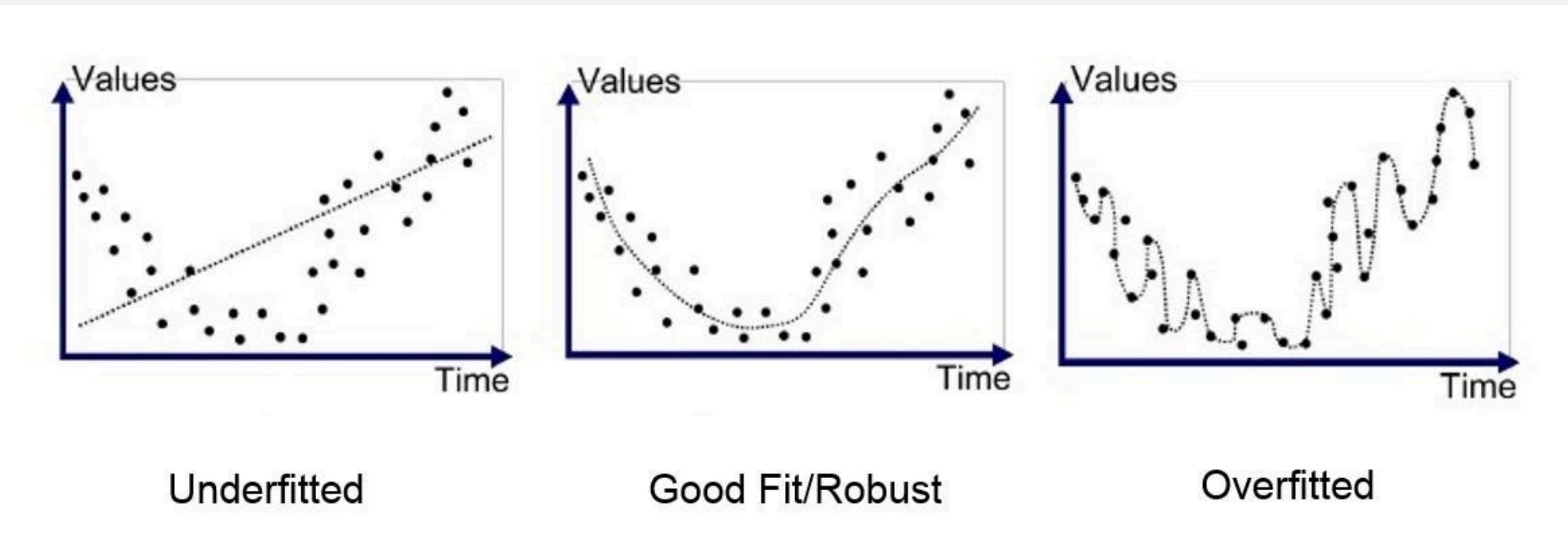


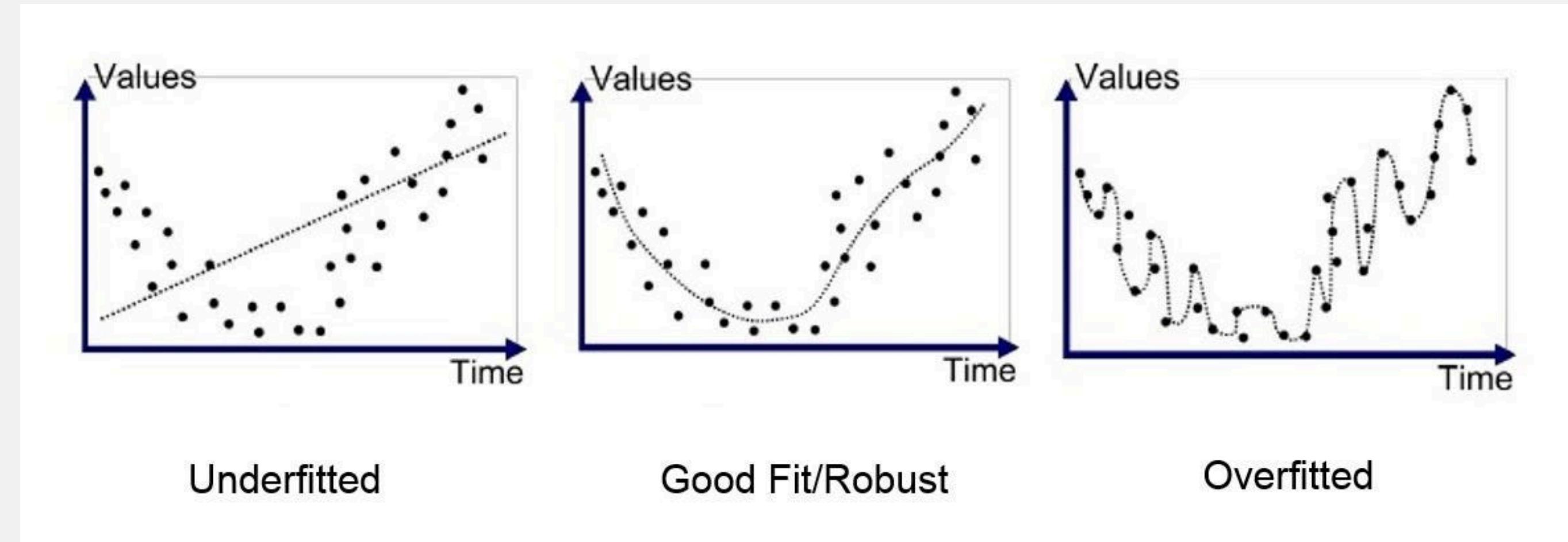
<https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>



Common Problem

Underfitting vs Overfitting





What it is	Too basic to capture patterns	Captures true underlying patterns	Memorizes training data, not general patterns
Example	Straight line for curved house prices	Student understands and applies concepts	Student memorizes pyp answers but fails on exam
Signs	Poor performance on both train/test	Good on both train/test	Great on training, bad on new data
Symptoms	High bias, low variance	Balanced bias and variance	Low bias, high variance
Solution	Use more complex model or features	Balance model complexity and data quality	Use simpler model, more data, or regularization

How to Determine?

Case 1:

Training accuracy: 60%
Test accuracy: 58%

low training accuracy
low test accuracy

= Underfitting

Case 2:

Training accuracy: 98%
Test accuracy: 65%

high training accuracy
low test accuracy

= Overfitting

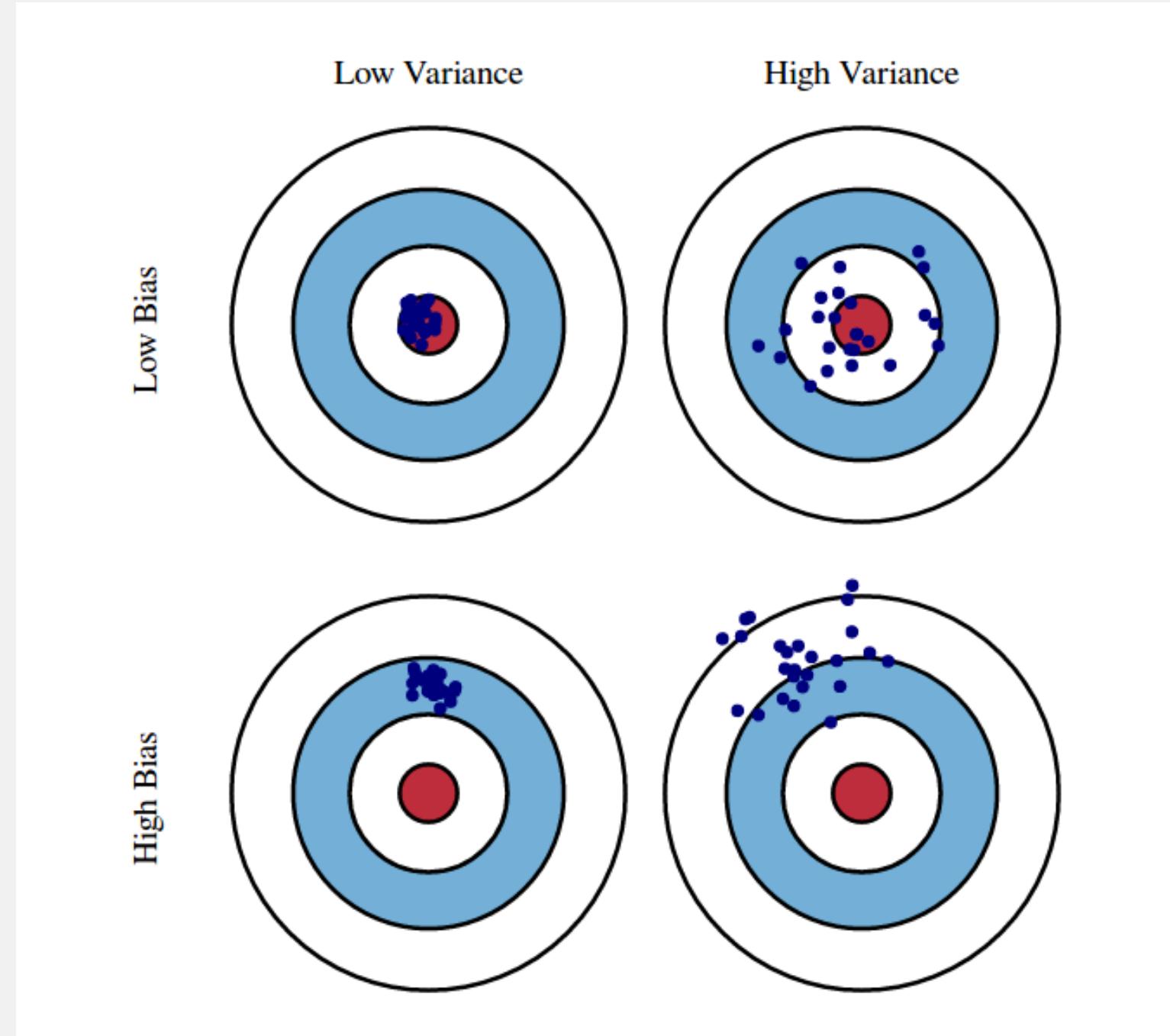
Case 3:

Training accuracy: 90%
Test accuracy: 90%

high training accuracy
high test accuracy

= Good fit 

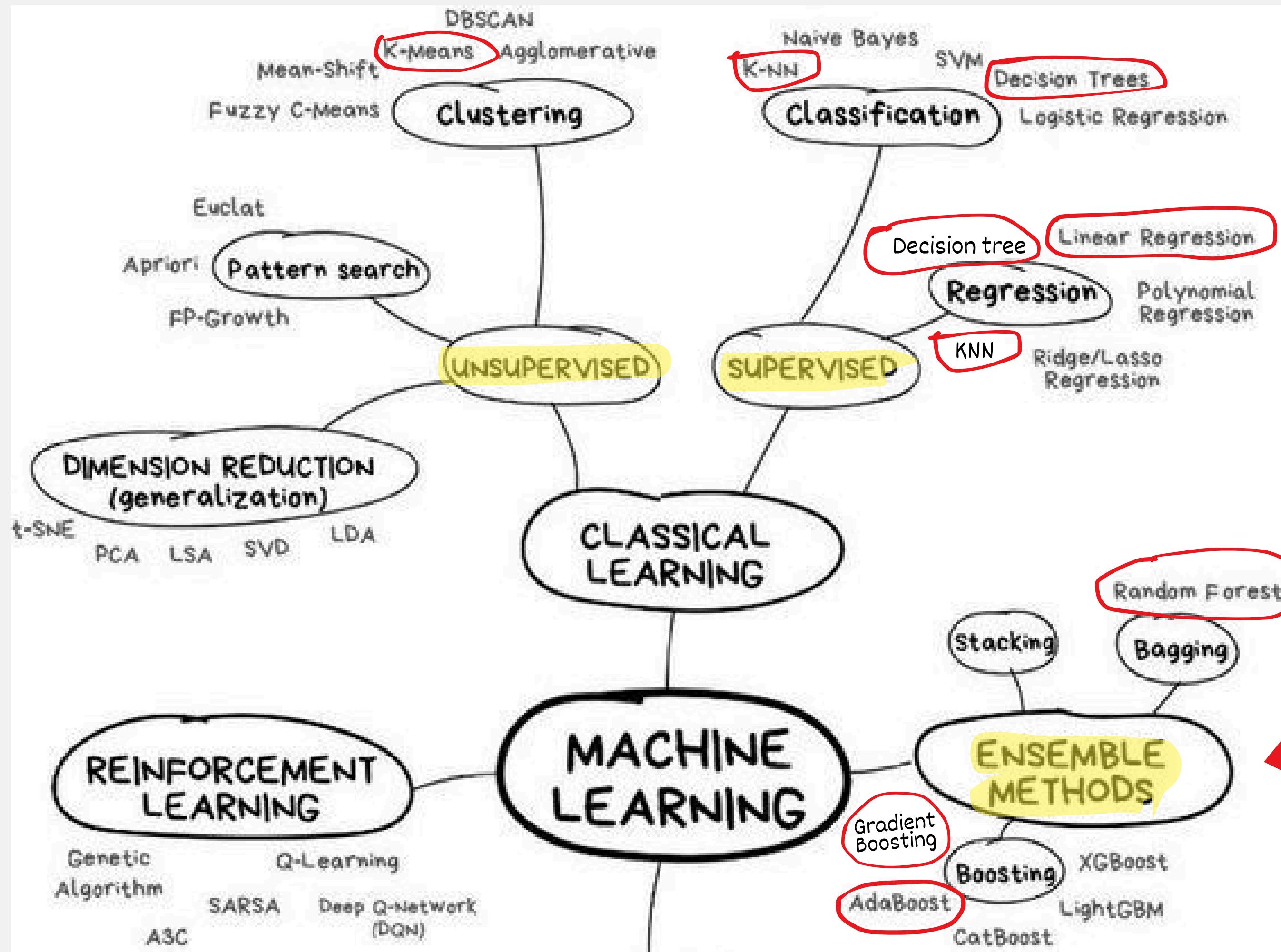
Bias-Variance Tradeoff



Bias: Error due to overly simple assumptions (underfitting)

Variance: Error due to too much complexity (overfitting)

Goal: Find a balance where both are low



Ensemble Methods

(Combine models to improve results)

⚙️ How It Works

- You train multiple models then combine their outputs by:
 - Averaging (for regression)
 - Voting (for classification)

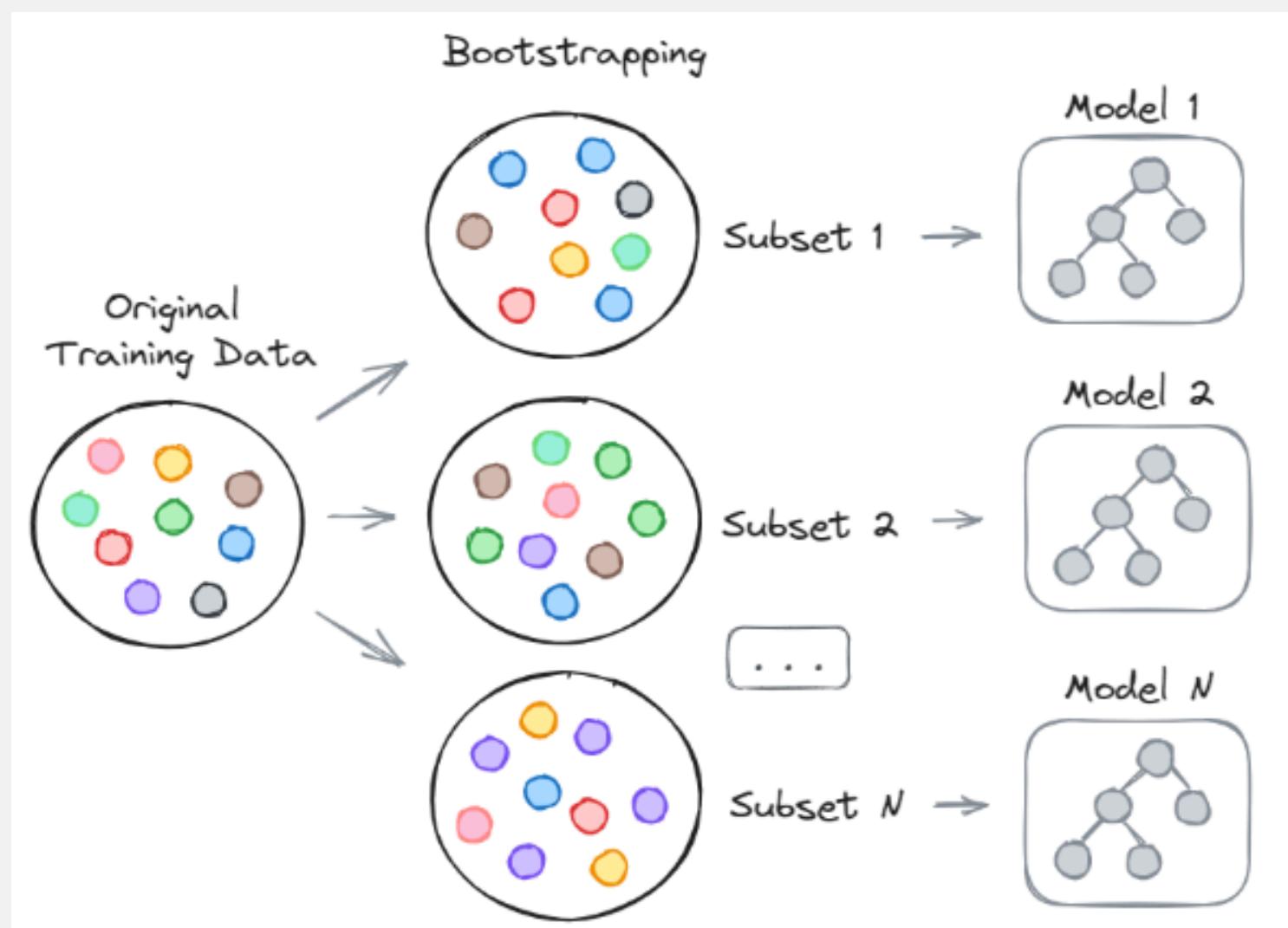


Types of Ensemble Methods

(Mainly used for tree)

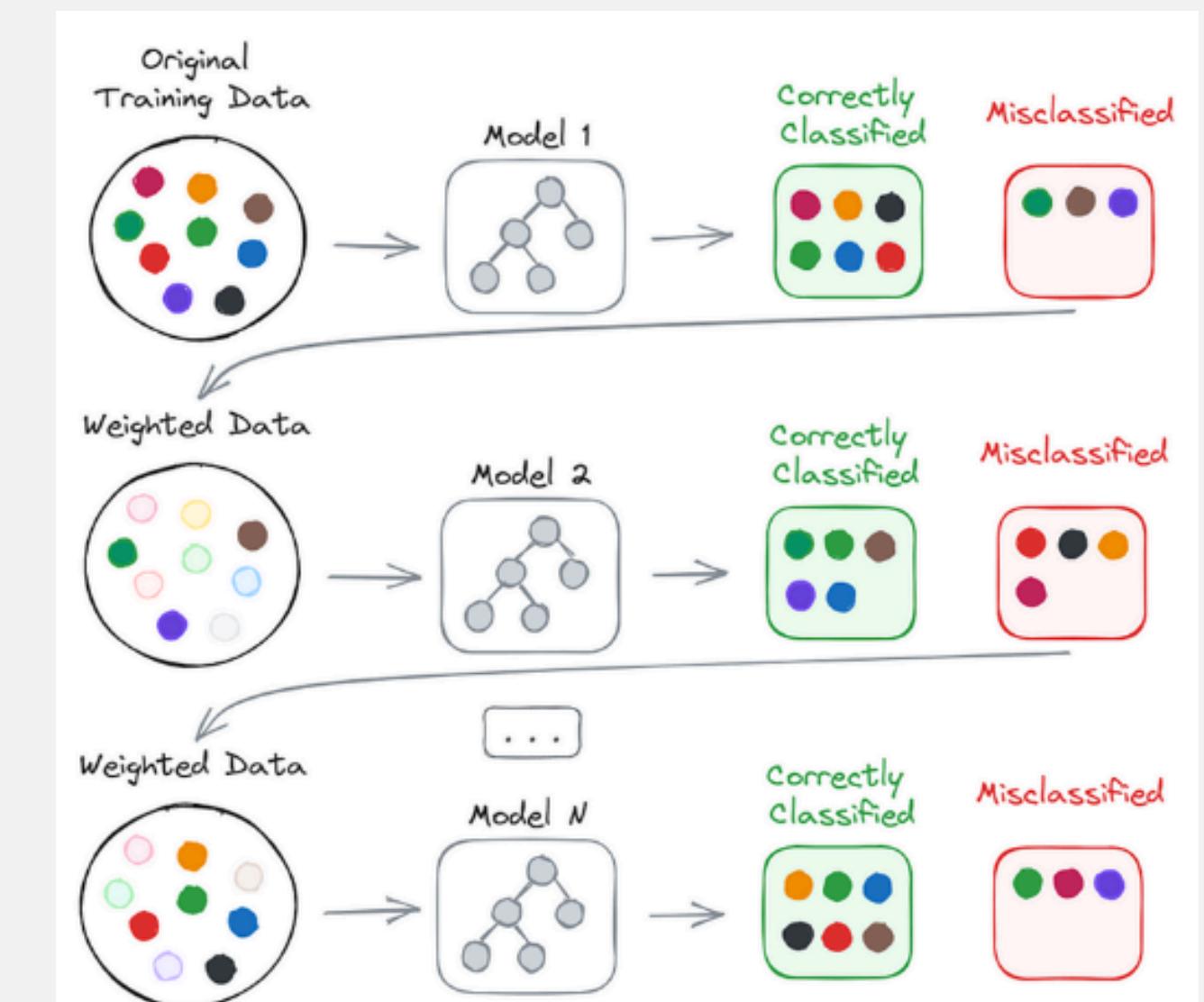
Bagging

Train on random subsets of the data in parallel
(Reduce variance)



Boosting

Train models sequentially to fix previous mistakes
(Reduce bias)



Bagging & Boosting

✓ Pros

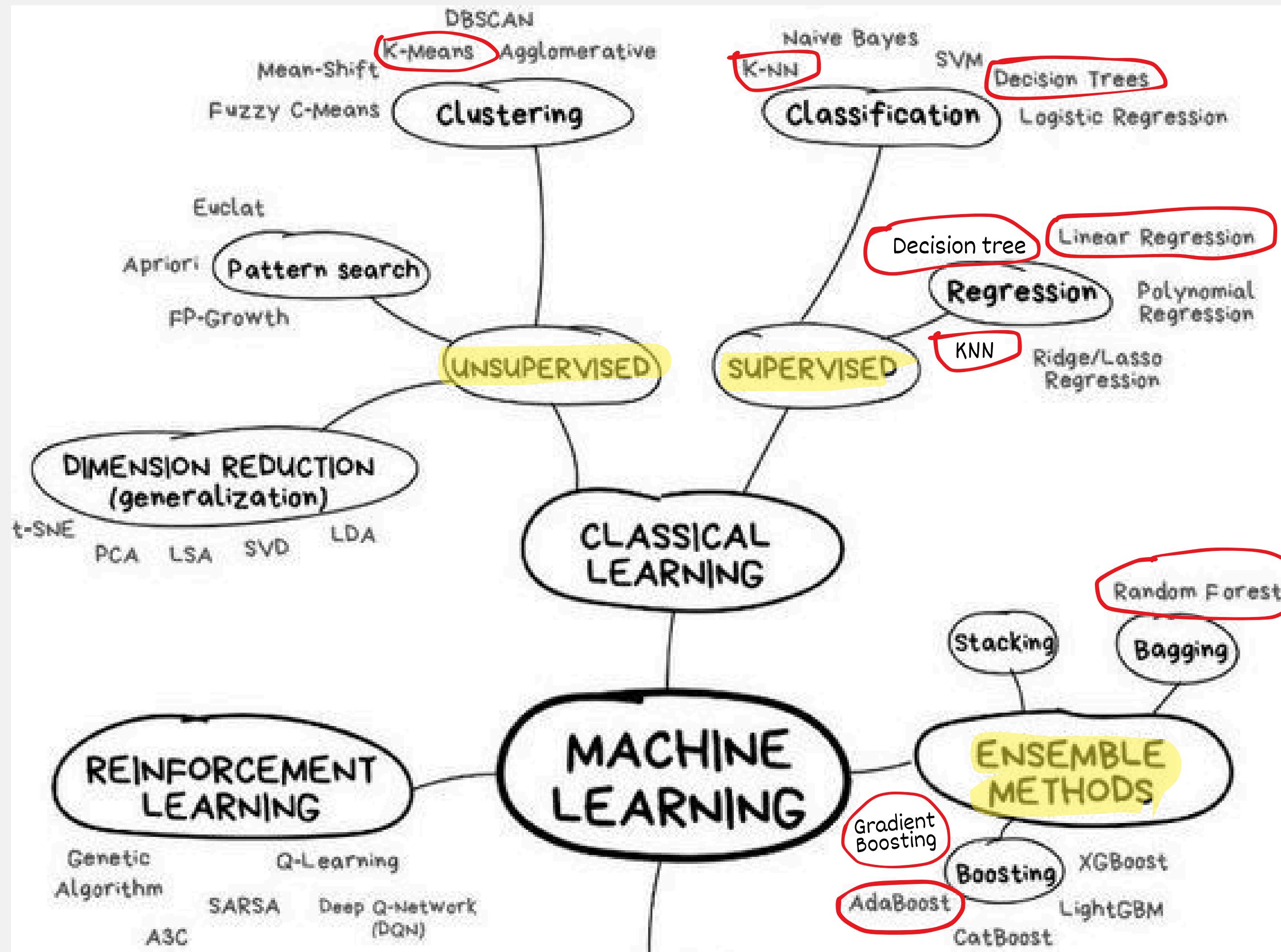
- Higher accuracy 
- Works well on complex data
- Reduces overfitting (esp. bagging)

✗ Cons

- More computationally expensive 
- Harder to interpret
- Risk of overfitting if not tuned properly



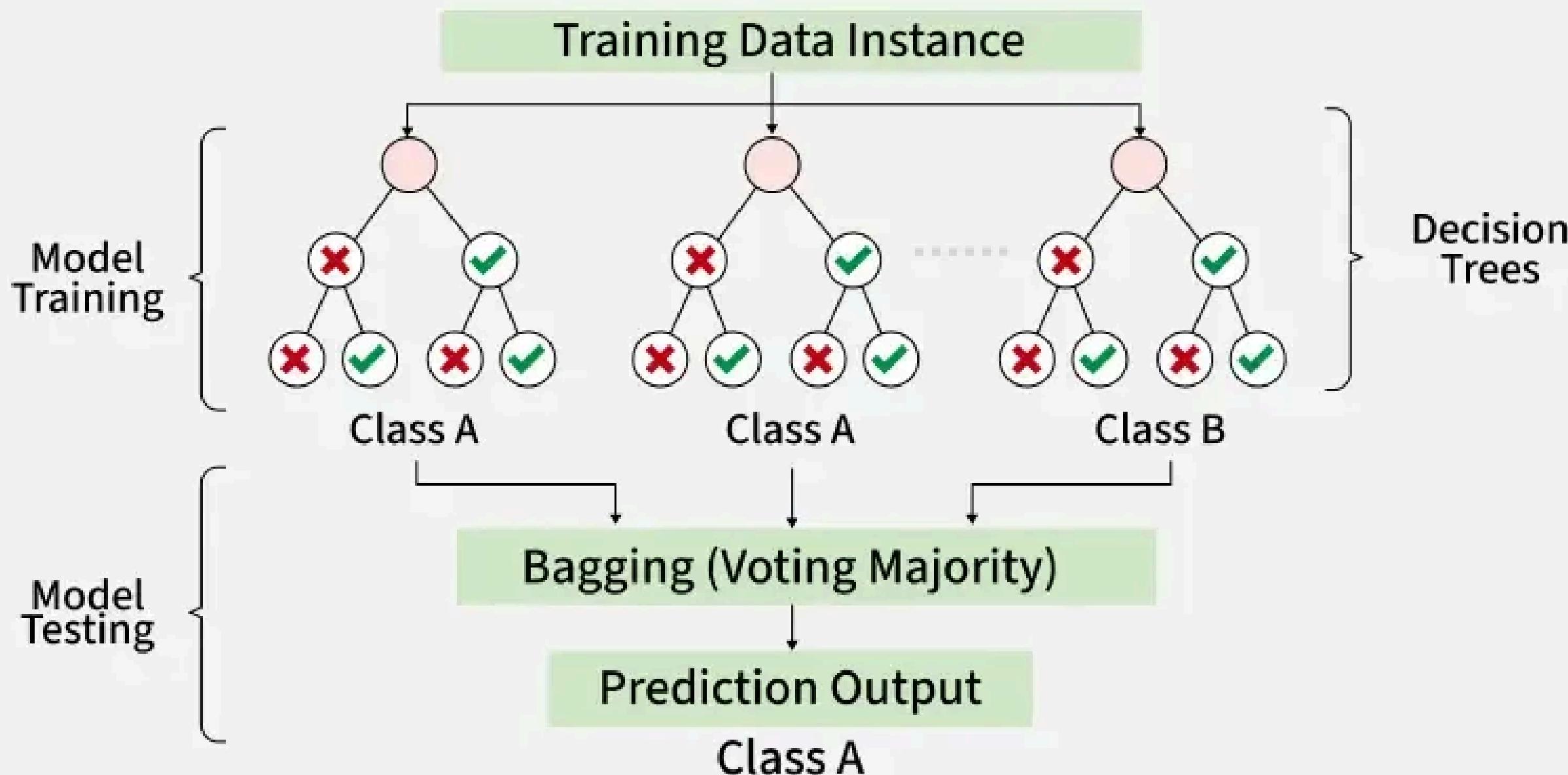
WE COVER EACH OTHER'S WEAKNESSES



Random Forest

(Bagging ensemble algorithms)

- A bunch of decision trees voting together



Random Forest

✓ Pros

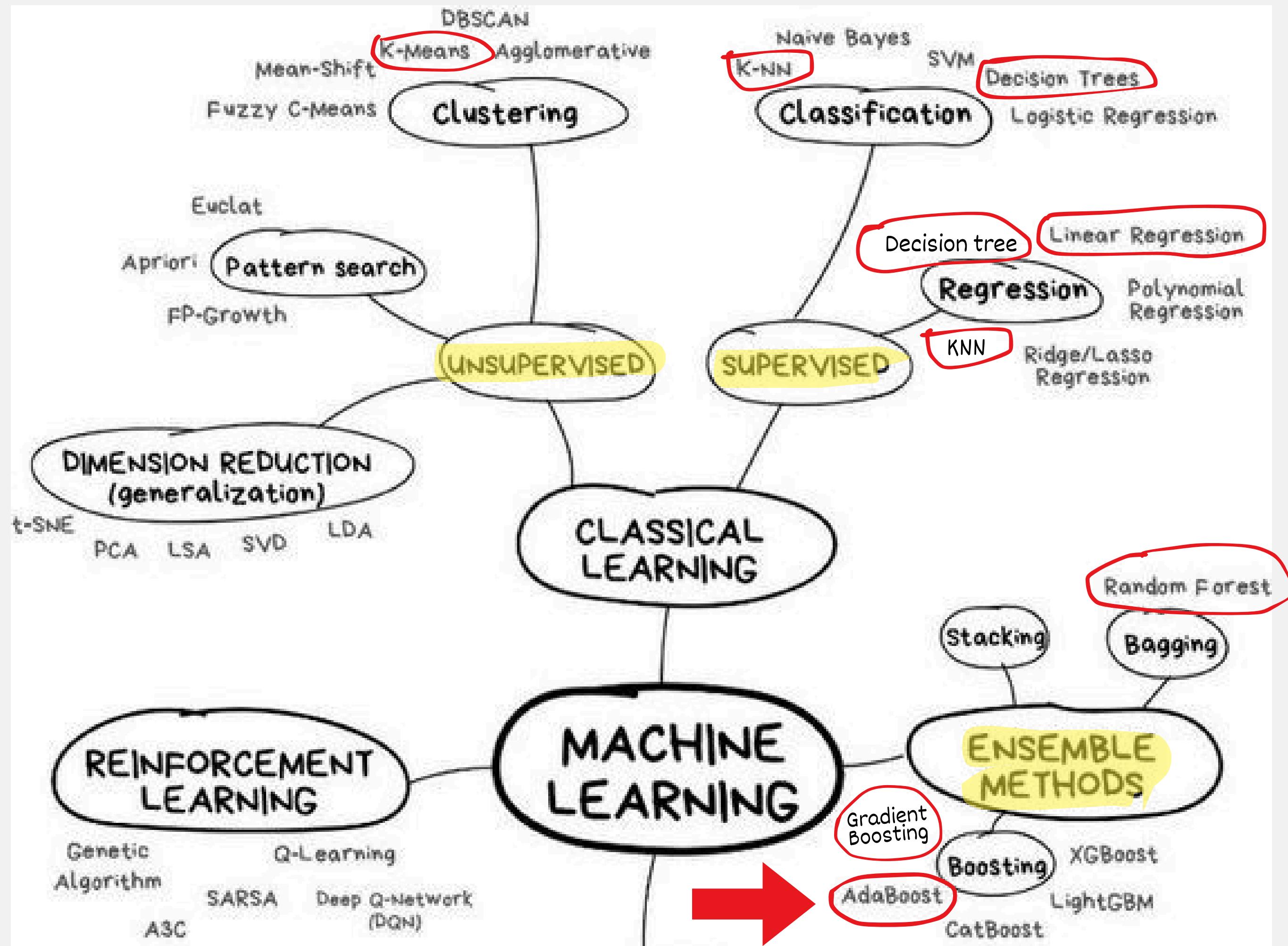
- Handles both regression & classification
- Great accuracy out of the box
- Reduces overfitting (thanks to averaging)
- Handles missing values & noisy data well

✗ Cons

- Slower to train with many trees
- Hard to interpret (black-box feel)
- Can be memory-heavy



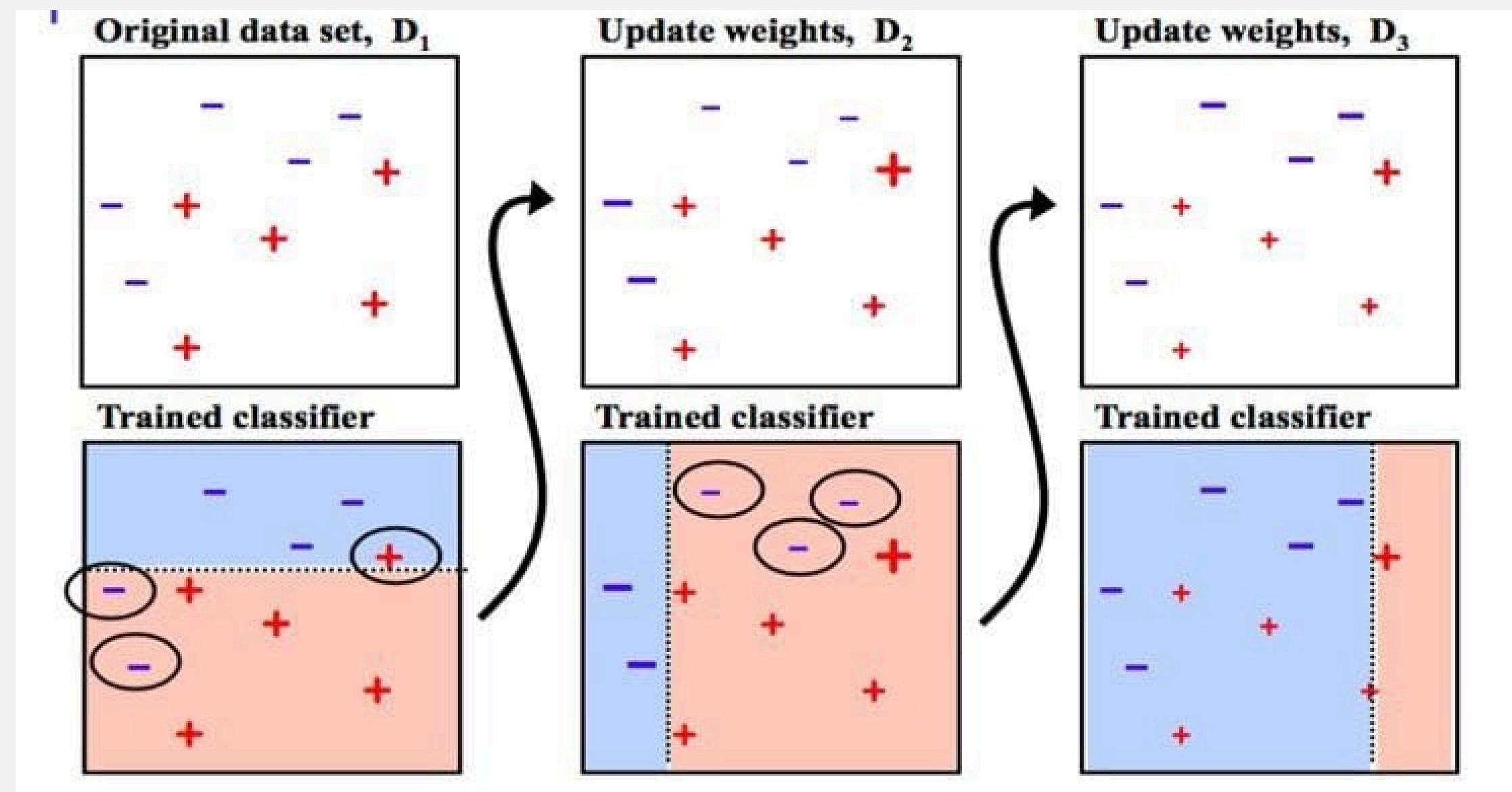
ONE TREE MIGHT BE WRONG, BUT TOGETHER
THEY'RE USUALLY RIGHT.



AdaBoost

(Boosting ensemble algorithms)

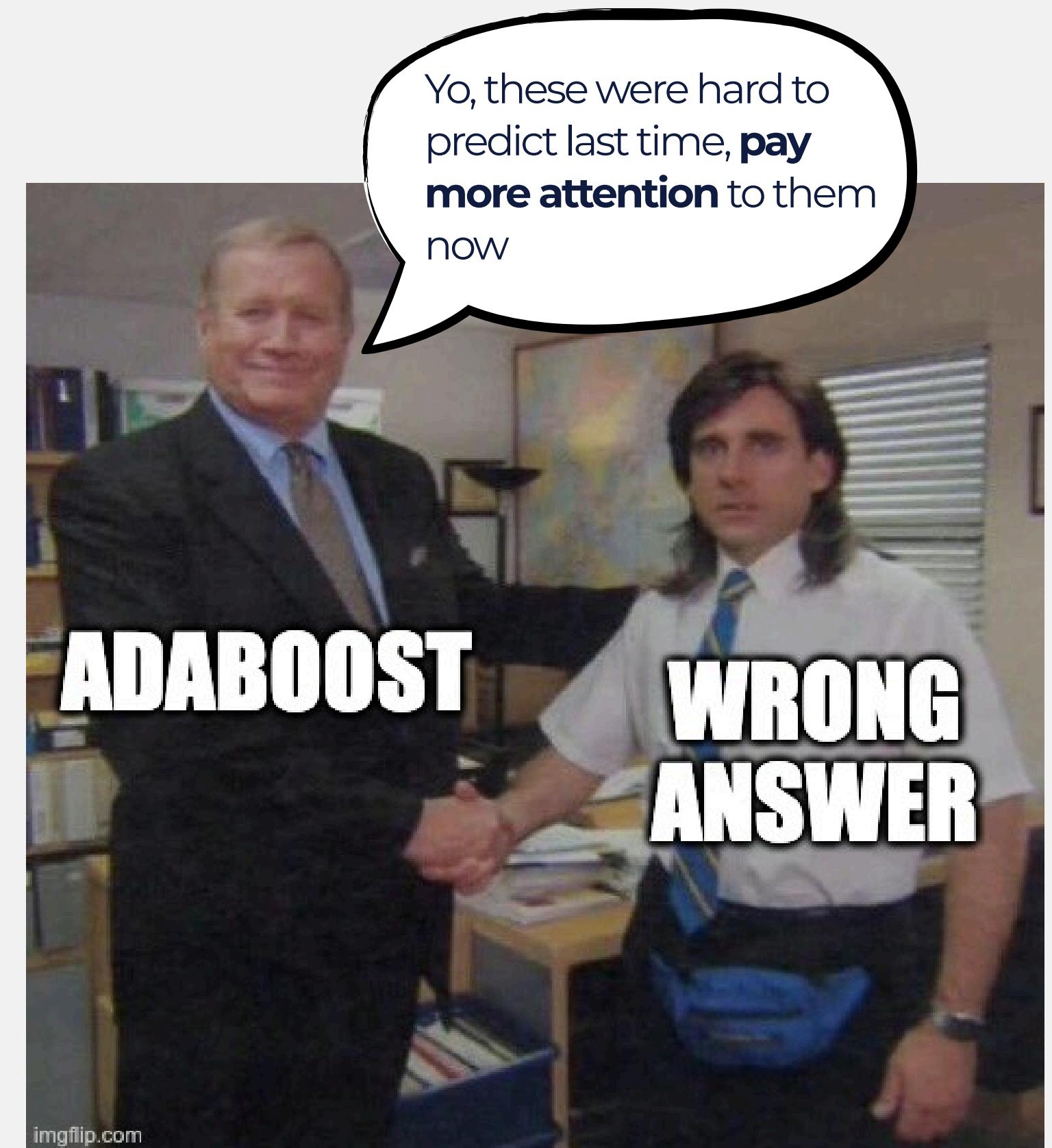
- Fix wrong classifications by increasing the weight on wrongly predicted samples

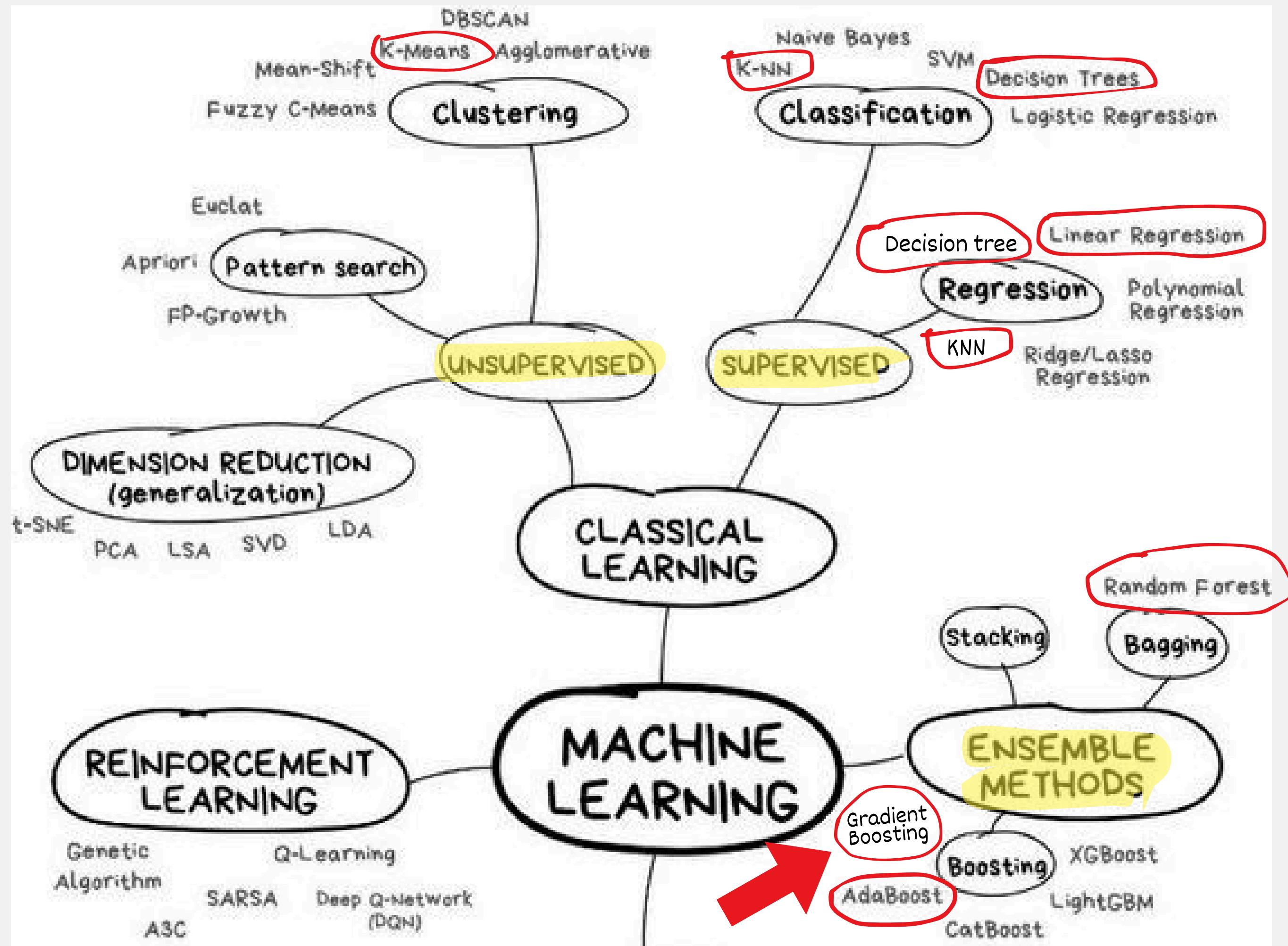


AdaBoost

(Boosting ensemble algorithms)

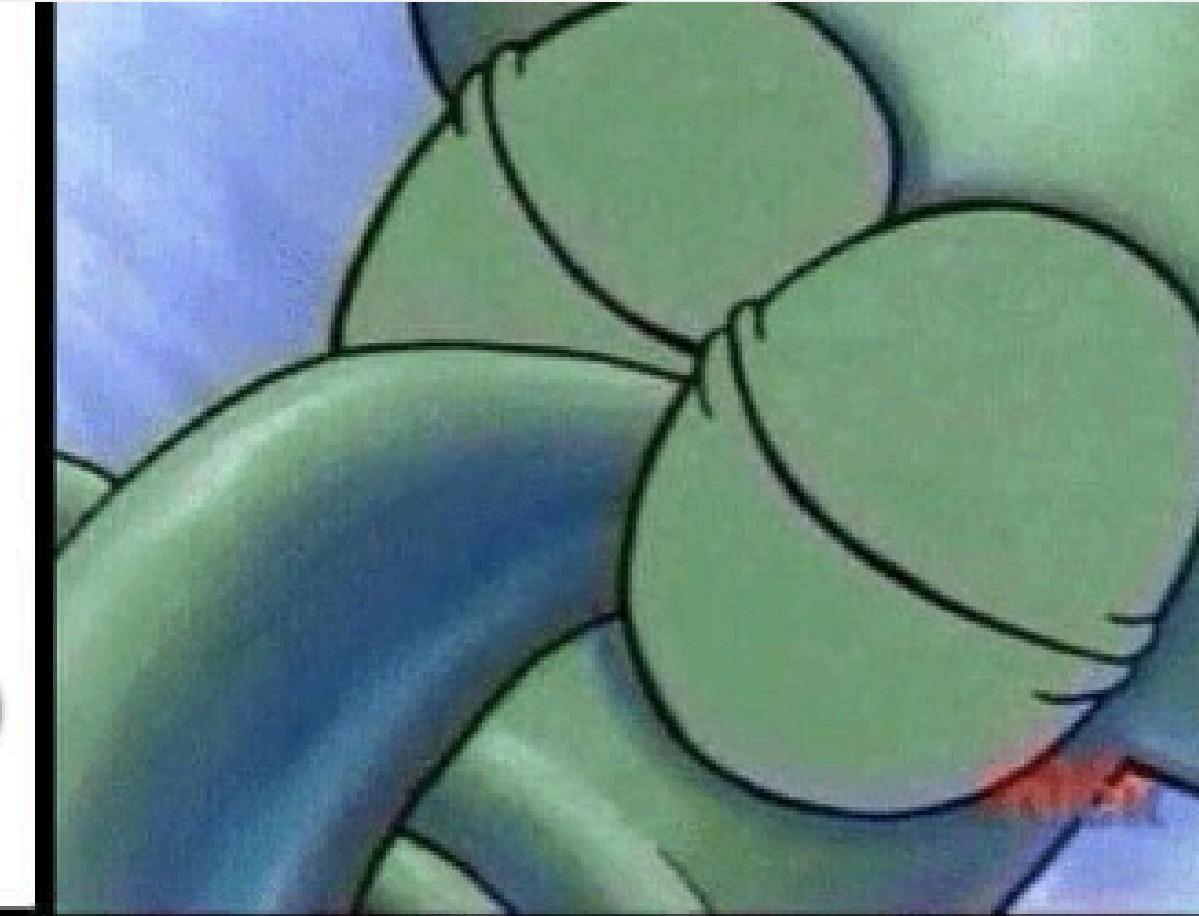
- Teacher re-testing students
 - each time, spending more time on the students who failed before





**LEARNING
THE
THEORY FOR ML**

**CODE
THE THING**



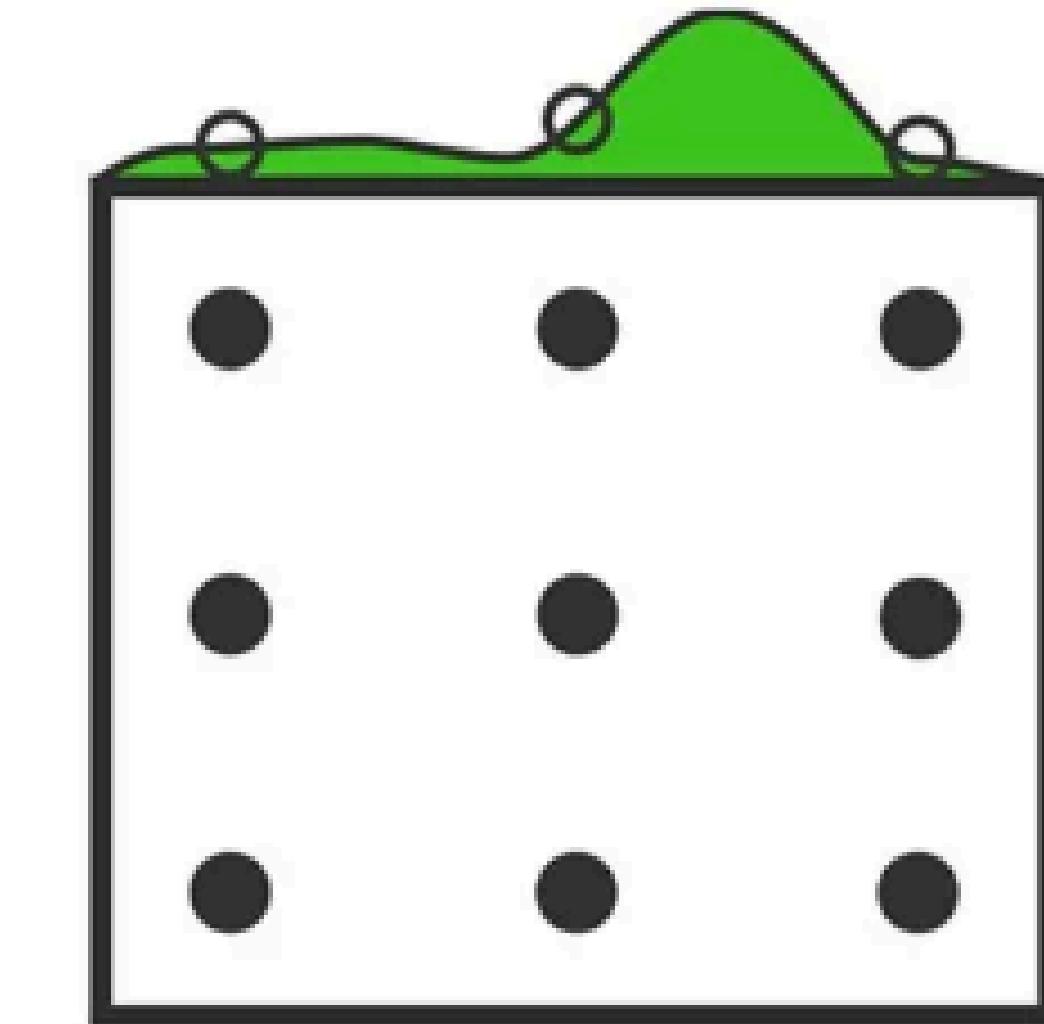


The Ultimate Guide to Hyperparameter Tuning | Grid Search vs. Randomized Search



HYPERPARAMETER TUNING

Hyperparameter 1



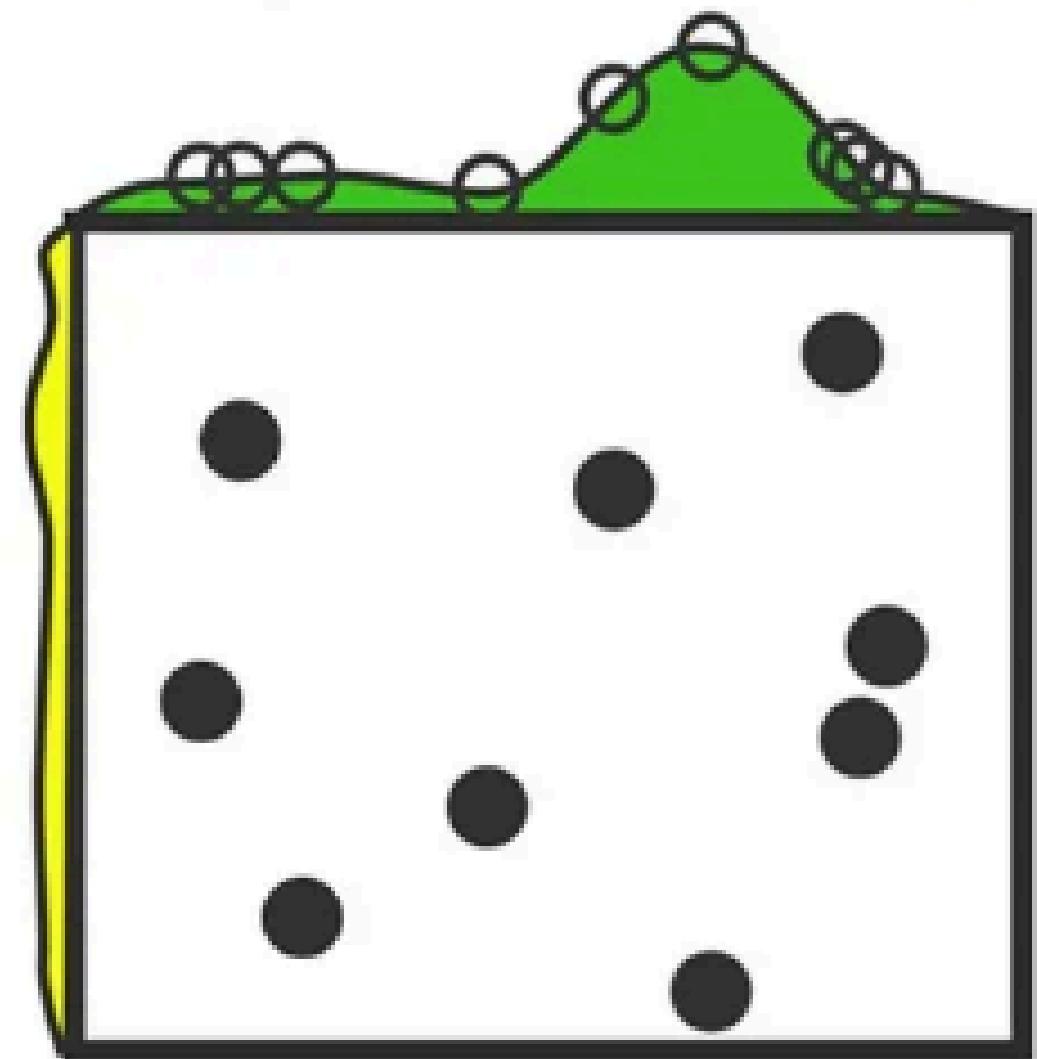
GRID SEARCH

Watch on YouTube



VS

Hyperparameter 2



RANDOMIZED

Hyperparameter 1