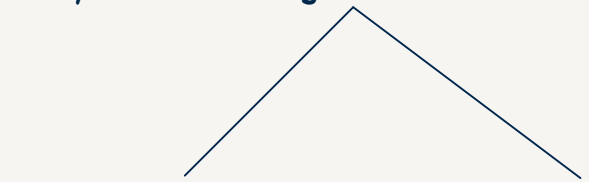
أكاديمية كاوست
KAUST ACADEMY

# Day 1: Machine Learning Foundations Summary

Note: Do Not depend entirely on this and study from the official slides.

Contributed by: Hassan Mohammed Nasr

- Machine learning (ML) is the field that gives the computers the ability to learn using data without being explicitly programmed.



**Structured data**

- Rows and columns
- Used in ML

**Unstructured data**

- Complex Format
- Used in DL

- ML types:
  - Supervised learning ⟶ Map x (input / features) to y (output / target / label)
    - Regression: continuous output
    - Classification: discrete output
  - UnSupervised learning ⟶ Find hidden structure in input X.
    - Clustering or grouping.
    - Dimensionality reduction
  - Reinforcement learning ⟶ Learn from trial and error (maximize reward ).

## • Three main pillars / components for any supervises learning:

### 1 - hypothesis (model)

- function that maps input to the prediction
- Works by updating the parameters (weights) to fit the data.
- example: linear regression

### 2 - Loss Function (Cost)

- differentiable function.
- measure the error.
- objective: minimize loss
- example: mean squared error (MSE)

### 1- hypothesis (model)

- algorithm that updates weights bases on the loss
- Closed Form: Find exact solution
- Iterative form: Requires iterations
  example: gradient descent

## • Gradient Descent

1- compute loss ⟶ 2- compute gradient (derivative) ⟶ 3- Update weights (take step in opposite direction) ⟶ 4- Repeat

# Problems in ML

## Bad Data

1- not enough data: Small Datsets
2- Non-representive data: data doesn't reflect the task
3-poor data: Noisy date and missing values
solution: data cleaning
4- irrelevant features: bad selection of features.
solution: feature engineering.

## Bad Algorithm

1- Overfitting:
- model is too complex and learns noise "memorize"
- high variance, low bias
- Very good in training, but bad in real life
- Train error low, test error high

2- Underfitting:
- model is too simple and learns nothing
- high bias, low variance
- bad in both training and real life
- Train error and test error low

- Our Goal is generalization: making the model behave well on unseen data

How to mimic?

1- split data to train and test
- The holdout Split ( train test split) → Problem when data is small
- K-Fold Split → Every data point serve as test sample once → Problem: Class imbalance
- Stratified Split → Solves imbalance problem

2- Evaluate using metrics:
- Metric → Interpretable by humans
- Loss → Must be differentiable (used by models not humans)

3- Mitigate and Fix:
- Overfitting → Reduce complexity, add data, remove noise, Do Regularization
- Underfitting → Increase complexity, add features

## • Some Metrics

1- Accuracy: The percentage of predictions that are correct → $\frac{\text{\# Correct predictions}}{\text{\# All predictions}}$ → Bad when imbalance

2- Precision: How often we predicted Yes as correct prediction → $\frac{\text{True Positive}}{\text{True positive + False positive}}$ →
"Be right when we say yes"
Used when false positive are costly
"Don't say yes unless very sure"

3- Recall: How many yes we can find → $\frac{\text{True Positive}}{\text{True positive + False negative}}$ →
False negatives are dangerous
"We want to catch as many yes as possible"
"Find all yes"

4- F1 Score: Balances recall and precision → $2 \times \frac{\text{Precision x Recall}}{\text{Precision + Recall}}$ → Used when Imbalance