

## Questions on ADABOOST

### Q1. Dataset

Use the **SMS Spam Collection Dataset**  
(Source: UCI / Kaggle → spam.csv)

#### Task

Classify SMS messages as:

- spam (1)
- ham (0) — not spam

#### Data Description

Column	Meaning
label	spam / ham
text	SMS message content

There are ~5,500 messages

#### Part A — Data Preprocessing & Exploration

1. Load the SMS spam dataset
2. Convert label: "spam" → 1, "ham" → 0
3. Text preprocessing:
  - Lowercase
  - Remove punctuation
  - Remove stopwords
4. Convert text to numeric feature vectors using **TF-IDF vectorizer**
5. Train-test split (80/20)
6. Show class distribution

#### Part B — Weak Learner Baseline

Train a **Decision Stump**:

`DecisionTreeClassifier(max_depth=1)`

Report:

- Train accuracy
- Test accuracy
- Confusion matrix
- Comment on why stump performance is weak on text data

#### Part C — Manual AdaBoost (T = 15 rounds)

Implement AdaBoost **from scratch** and after **each iteration, print**:

- Iteration number

- Misclassified sample indices
- Weights of misclassified samples
- Alpha value

Then update and normalize weights.

Also produce:

- Plot: iteration vs weighted error
- Plot: iteration vs alpha

Finally report:

- Train accuracy
- Test accuracy
- Confusion matrix
- Short interpretation of weight evolution

## Part D — Sklearn AdaBoost

Train:

```
AdaBoostClassifier(
    base_estimator=DecisionTreeClassifier(max_depth=1),
    n_estimators=100,
    learning_rate=0.6
)
```

Report:

- Train accuracy
- Test accuracy
- Confusion matrix
- Compare performance with manual implementation

## Q2. Dataset Description

You will use the **UCI Heart Disease dataset** (available in `sklearn.datasets`). This dataset contains patient medical features used to predict heart disease.

Feature	Meaning
Age	Patient age
Sex	Gender (1 = male, 0 = female)
Cp	Chest pain type (0–3)
Trestbps	Resting blood pressure
Chol	Serum cholesterol (mg/dl)
Fbs	Fasting blood sugar >120 mg/dl (1/0)
Restecg	Resting ECG results
Thalach	Max heart rate achieved
Exang	Exercise-induced angina (1/0)
Oldpeak	ST depression induced by exercise
Slope	Slope of peak exercise ST segment
Ca	# of major vessels (0–3)
Thal	Thallium stress test result (0–3)

**Target:**

1 = heart disease present

0 = No heart disease

**Part A — Baseline Model (Weak Learner)**

1. Load the dataset and preprocess (handle categorical features, scaling if needed)
2. Train **one Decision Stump** (`max_depth = 1`)
3. Report:
  - Training & test accuracy
  - Confusion matrix
  - Classification report
4. What shortcomings do you observe in a single stump?

**Part B — Train AdaBoost**

1. Train **AdaBoostClassifier** using decision stumps as base learners
2. Use:
  - `n_estimators = [5, 10, 25, 50, 100]`
  - `learning_rate = [0.1, 0.5, 1.0]`
3. For each combination:
  - Train model
  - Compute accuracy on test set
4. Plot:
  - `n_estimators` vs accuracy for each `learning_rate`
5. Identify best config (highest accuracy)

**Part C — Misclassification Pattern**

1. For the **best model**, collect the **sample weights** and **prediction errors** at each iteration.
2. Plot:
  - Weak learner error vs iteration
  - Sample weight distribution after final boosting stage
3. Explain:
  - Which samples got highest weights?
  - Why does AdaBoost focus on them?

**Part D — Visual Explainability**

1. Plot feature importance from AdaBoost.
2. Identify top 5 most important features.
3. Explain why these features may matter medically.

**Q2. Dataset:****WISDM Smartphone & Watch Motion Sensor Dataset**

(Available on UCI / Kaggle → `WISDM_ar_v1.1_raw.txt`)

**Dataset Description**

Collected from smartphones & smartwatches using **accelerometer & gyroscope**.

Attribute	Description
<code>user_id</code>	Person ID
<code>Activity</code>	type of physical activity (e.g., walking, jogging, sitting)

Timestamp	time in milliseconds
sensor readings	acceleration or gyroscope X, Y, Z values

Target prediction task:

- Convert activity into **binary label**:
  - **1** = vigorous motion (Jogging, Upstairs)
  - **0** = light/static motion (Walking, Sitting, Standing, Downstairs)

We will use only **accelerometer features** (X, Y, Z).

## Goal

Build an activity classifier using **AdaBoost** to distinguish between **vigorous vs normal activity** based on smartphone sensor accelerations.

## Part A — Data Preparation

1. Load the dataset (WISDM\_ar\_v1.1\_raw.txt)
2. Extract only numeric accelerometer X, Y, Z columns
3. Create activity label as binary:

Activity Type	Label
Jogging, Up	1
Walk, Sit, Stand, Down	0

4. Handle missing/dirty entries
5. Train-test split (70/30)

## Part B — Weak Classifier Baseline

Train a **Decision Stump**:

`DecisionTreeClassifier(max_depth=1)`

Report:

- Accuracy (train + test)
- Confusion matrix
- Interpretation of stump result

## Part C — Manual AdaBoost (T = 20 rounds)

Write your own AdaBoost **with full weight tracking**.

At **each iteration**, print:

- Iteration number
- Misclassified sample indices
- Weights of misclassified samples

**Note:** Normalize weights after update.

Also plot:

- Boosting round vs error
- Boosting round vs alpha

Finally report:

- Train accuracy
- Test accuracy
- Confusion matrix
- Interpretation: how weights shifted over time

## **Part D — Sklearn AdaBoost**

Train:

```
AdaBoostClassifier(  
    base_estimator = DecisionTreeClassifier(max_depth=1),  
    n_estimators = 100,  
    learning_rate = 1.0  
)
```

Report:

- Train/Test accuracy
- Confusion matrix
- Compare with your manual implementation