# Lab Experiment: Exploring Hugging Face – From Manual ML to Pretrained Models

## Course Outcome Focus

- Understand ML concepts manually
- Implement ML using code
- Explore modern ML using pretrained models via **Hugging Face**

## Question

You are given a small dataset for a **binary classification problem** (e.g., Spam Detection or Sentiment Analysis).

### Task 1: Build a Model from Scratch

1. Choose a suitable **machine learning algorithm** (Decision Tree / Naïve Bayes / Logistic Regression).
2. Solve the problem **manually** using a dataset of **minimum 10 data points**.
3. Implement the same model using **Python code from scratch** (without using pretrained models).

### Task 2: Solve the Same Problem Using a Pretrained Model

1. Use a **pretrained model from Hugging Face** to solve the same classification problem.
2. Compare the results with the manually solved and coded model.
3. Comment on performance, ease of use, and accuracy.

# Sample Dataset (10 Data Points – Sentiment Analysis)

| ID | Text | Sentiment |
|---|---|---|
| 1 | I love this product | Positive |
| 2 | Very bad experience | Negative |
| 3 | Amazing quality | Positive |
| 4 | Not worth the money | Negative |
| 5 | Excellent performance | Positive |
| 6 | Terrible support | Negative |
| 7 | Happy with purchase | Positive |
| 8 | Waste of time | Negative |
| 9 | Good value | Positive |
| 10 | Disappointed | Negative |

## ◇ Part A: Manual Solution (Conceptual) – This has to be in Observation

**Chosen Algorithm:** Naïve Bayes (Text Classification)

Steps (Manual):

1. Count word frequencies for **Positive** and **Negative** classes.
2. Compute prior probabilities:
    - P(Positive) = 5/10
    - P(Negative) = 5/10
3. For a new sentence *"Excellent product"*:
    - Compute likelihood of words under both classes.
    - Multiply likelihood with prior.
4. Assign class with **maximum posterior probability**.

**Manual Result:**
Sentence classified as **Positive**

## ◇ Part B: Code-Based Implementation (From Scratch)

**Approach:**

- Convert text to bag-of-words
- Train Naïve Bayes using `scikit-learn` (no pretrained models)

**Outcome:**

- Model correctly classifies most test samples
- Accuracy depends heavily on dataset size

## ◇ Part C: Pretrained Model using Hugging Face

**Steps:**

1. Load a pretrained sentiment model from Hugging Face
   Example: `distilbert-base-uncased-finetuned-sst-2-english`
2. Use Hugging Face `pipeline("sentiment-analysis")`
3. Input same text samples
4. Observe predictions

**Output Example:**

```
Text: "Excellent product"
Prediction: POSITIVE (Score: 0.99)
```