# Lab #5 : Ensemble Learning

*Objectives* :

- Understand and implement ensemble learning techniques: Bagging, Random Forest, AdaBoost, and Stacking.
- Compare the performance of individual models versus ensemble models.
- Gain practical experience in handling a real-world, imbalanced dataset.

## Dataset : Use Iris dataset

## Lab Instructions

### Step 1: Data Preparation

1. Load the dataset into Python using pandas.
2. Explore the dataset:
   - Check for missing values.
   - Identify categorical and numerical features.
   - Check for class imbalance (especially in fraud detection dataset).
3. Preprocess the data:
   - Encode categorical features (e.g., One-Hot Encoding or Label Encoding).
   - Scale numeric features (StandardScaler or MinMaxScaler).
4. Split the dataset into **training** and **testing** sets (e.g., 80% train, 20% test).

### Step 2: Train Individual Models

5. Select **at least two baseline classifiers** such as:
   - Decision Tree
   - K-Nearest Neighbors
   - Logistic Regression
6. Train the models on the training set.
7. Evaluate their performance on the test set using:
   - Accuracy
   - Precision, Recall, F1-Score
   - Confusion Matrix

### Step 3: Bagging

8. Implement **Bagging** using a base estimator (e.g., Decision Tree).
9. Experiment with the **number of estimators** (e.g., 50, 100).
10. Evaluate the performance using the same metrics as above.
11. Compare results with baseline models.

## Step 4: Random Forest

12. Train a **Random Forest** classifier.
13. Experiment with **number of trees** and **max depth**.
14. Evaluate performance.
15. Discuss why Random Forest is more stable than a single Decision Tree.

## Step 5: Boosting

16. Implement **AdaBoost** using a weak learner (e.g., Decision Tree with depth 1).
17. Experiment with **number of estimators** and **learning rate**.
18. Evaluate and compare performance with Bagging and Random Forest.

## Step 6: Stacking

19. Choose **at least two different classifiers** as base learners (e.g., Decision Tree, KNN, Logistic Regression).
20. Select a **meta-learner** (e.g., Logistic Regression).
21. Train the stacking ensemble.
22. Evaluate the performance.
23. Compare results with Bagging, Random Forest, and Boosting.

## Step 7: Performance Comparison

24. Collect metrics for all models:
    - Accuracy, Precision, Recall, F1-Score
25. Discuss:
    - Which ensemble performed best?
    - Which hyperparameters had the most impact?

## Step 8: Optional Extensions

- Try **VotingClassifier** (hard vs soft voting).
- Experiment with **feature selection** before applying ensemble methods.
- Handle **class imbalance** using techniques like **SMOTE** or **class weighting**.
- Test ensembles on a different dataset (e.g., Adult Income vs Credit Card Fraud).