# Mohammad Bagher Soroush - 400130273 - ML 5 Report

**1 .Explanation of Process Steps**

Dataset Loading:

- The dataset was loaded from a CSV file using pandas. Unnecessary columns (Unnamed: 0 and Loan_ID) were dropped as they were identifiers not useful for prediction.

Preprocessing:

- The target column, Loan_status_Y, was separated from the features.
- The Dependents column was cleaned by replacing '3+' with 3 and converting it to integers.
- Missing values in numerical columns were filled with their mean, while categorical columns were filled with their mode.

Model Training:

- The dataset was split into training and testing sets.
- Three machine learning models were implemented: Random Forest (Bagging), AdaBoost, and Gradient Boosting.
- Each model was trained using the training data and evaluated using accuracy on the test data.
- Model Comparison: The accuracies of all three models were printed for comparison.

**2. Bias and Variance**

- Bias: The error introduced by simplifying the model too much, causing it to miss the relevant patterns in the data (underfitting).
- Variance: The error introduced by the model's sensitivity to small fluctuations in the training data (overfitting).
- A model must strike a balance between bias and variance for optimal performance.

**3 .Bagging vs. Boosting**

Bagging:

- Stands for Bootstrap Aggregating.
- Trains multiple independent models on random subsets of the data (with replacement).
- Combines their outputs (e.g., via majority vote) to reduce variance.
- Example: Random Forest.
- Tends to reduce variance but does not significantly impact bias.

Boosting:

- Sequentially trains models, each correcting the errors of the previous ones.
- Focuses more on difficult-to-predict examples in subsequent iterations.
- Example: AdaBoost and Gradient Boosting.
- Tends to reduce bias, though it can lead to higher variance if overfitted.

**4 &5. Explanation of Models**

Random Forest (BaggingClassifier):

- Trains multiple decision trees on random subsets of the data.
- Combines their predictions to improve robustness and reduce overfitting.

Hyperparameters:

- n_estimators: Number of trees in the forest.
- max_depth: Maximum depth of each tree.
- max_features: Number of features considered at each split.

AdaBoostClassifier:

- Builds a sequence of weak learners (e.g., decision stumps), each focusing on correcting the errors of the previous ones.
- Combines all weak learners into a strong model.

Hyperparameters:

- n_estimators: Number of weak learners.
- learning_rate: Controls the contribution of each weak learner.
- base_estimator: The type of weak learner used.

GradientBoostingClassifier:

- Sequentially builds models by optimizing a loss function (e.g., log-loss) using gradient descent.
- Each model focuses on the residual errors of the previous models.

Hyperparameters:

- n_estimators: Number of boosting stages.
- learning_rate: Scales the contribution of each tree.
- max_depth: Maximum depth of the individual trees.