# Loan Approval Prediction via KNN and Decision Tree

---

## 1. Objective

Automate loan approval decisions by building and comparing two classification models—**K-Nearest Neighbors (KNN)** and **Decision Tree**—to predict a target **loan_status**.

---

## 2. Dataset Overview

| Column | Type | Description |
|---|---|---|
| `loan_id` | Integer | Unique application ID (to be dropped) |
| `no_of_dependents` | Integer | Number of dependents |
| `education` | Categorical | ‘Graduate’ / ‘NotGraduate’ |
| `self_employed` | Categorical | ‘Yes’ / ‘No’ |
| `income_annum` | Numeric | Annual income |
| `loan_amount` | Numeric | Requested loan amount |
| `loan_term` | Numeric | Repayment period (months) |
| `cibil_score` | Numeric | Credit score (higher = better) |
| `residential_assets_value` | Numeric | Value of residential assets |
| `commercial_assets_value` | Numeric | Value of commercial assets |
| `luxury_assets_value` | Numeric | Value of luxury assets |
| `bank_asset_value` | Numeric | Bank account balances |
| `loan_status` | Categorical | Target: Approved/ Rejected/ Not Known |

---

## 3. Data Cleaning & Preprocessing

1. **Drop Identifier**
   - Drop `loan_id` column from dataset.
2. **Handle Missing and Duplicate Values**
   - Drop rows with any missing values.
   - Drop duplicate rows if any.
   - Drop rows where value of ‘loan_status’ is **Not Known**
3. **Outlier Removal from ‘bank_asset_value’ column**
   - Detect and remove outliers using the IQR method.
4. **Feature Transformation (without using get_dummies)**
   - Convert `education` column from categoric to numeric: 1 for 'Graduate', 0 for ‘NotGraduate’. (Without using get_dummies)
   - Convert `self_employed` column from categoric to numeric: 1 for 'Yes', 0 for ‘No’. (Without using get_dummies)
5. **Feature–Target Split & Data Partition**
   - **Features (X):** all columns except `loan_status`

- **Target (Y):** `loan_status`
- **Train/Test Split:** 80% train, 20% test with `random_state=42`

---

## 4. Modeling Approaches

### 4.1 K-Nearest Neighbors (KNN)

- **Algorithm:** Classifies a sample based on the majority label among its **k** nearest neighbors in feature space.
- **Hyperparameter:** `n_neighbors=47`

### 4.2 Decision Tree

- **Algorithm:** Splits feature space on conditions that maximize information gain (using entropy).
- **Hyperparameter:** `max_depth=12`

---

## 5. Model Training & Evaluation

1. **Train** both models using the same train-test split on training set.
2. **Predict** outcomes on the test set for both models (knn and decision tree).
3. **Find and print** the following for both models:
    - **Accuracy**
    - **Confusion Matrix** (TP, TN, FP, FN)
    - **Sensitivity**
    - **Specificity**
4. **Area Plot Visualization** for both models:
    - Create a Stack Area Plot comparing **accuracy**, **sensitivity**, and **specificity** for KNN and Decision Tree classifiers as shown below.