

# *Predicting Loan Approval*

## *using Logistic Regression and Decision Trees*

### *with Regularization and Hyperparameter Tuning*

#### **Objective:**

The goal of this project is to build machine learning models that can predict whether a customer's loan application (or insurance claim) will be approved. The project focuses on comparing two popular algorithms – **Logistic Regression** and **Decision Tree Classifier** – while also applying **regularization techniques** and **hyperparameter optimization** (with GridSearchCV).

#### **Dataset:**

You can use a publicly available dataset from Kaggle, for example:

- **Loan Prediction Dataset:** Loan Prediction Problem Dataset on Kaggle  
<https://www.kaggle.com/datasets/ninzaami/loan-predication/data>

Column Name	Meaning	Explanation (Simple)	Example
Loan_ID	Loan Identifier	Unique ID for each loan application	LP001002
Gender	Applicant's Gender	Whether the applicant is Male or Female	Male, Female
Married	Marital Status	Whether the applicant is married or not	Yes, No
Dependents	Number of Dependents	Number of people dependent on the applicant (kids, family)	0, 1, 2, 3+
Education	Education Level	Applicant's education background	Graduate, Not Graduate
Self_Employed	Employment Type	Whether the applicant is self-employed or not	Yes, No
ApplicantIncome	Applicant's Income	Monthly income of the main applicant	5000
CoapplicantIncome	Co-applicant's Income	Monthly income of co-applicant (e.g., spouse)	2000
LoanAmount	Loan Amount	Amount of loan applied (in thousands)	128 = 128,000

<b>Loan_Amount_Term</b>	Loan Term	Time period of the loan (in months)	360 months = 30 years
<b>Credit_History</b>	Credit History	Record of applicant's past credit repayment (1 = good, 0 = bad)	1, 0
<b>Property_Area</b>	Property Location	Area type where the property is located	Urban, Semiurban, Rural
<b>Loan_Status</b>	Target Variable	Whether the loan was approved or not	Y = Approved, N = Not Approved

## Notes

- **Loan\_ID** is just an identifier → drop it before training.
- **Categorical variables** (Gender, Married, Education, Self\_Employed, Property\_Area, Loan\_Status) → need **encoding** (e.g., OneHotEncoder or LabelEncoder).
- **Dependents** looks numeric, but it's actually categorical because of "3+".
- **Credit\_History** is very important (strong predictor).
- **ApplicantIncome + CoapplicantIncome** can be combined into a **TotalIncome** feature.
- **LoanAmount** often has missing values → need to handle them (mean/median imputation or log-transform).
- **Loan\_Amount\_Term** is usually 360 months (30 years), but some records are shorter/longer.

### Real-world interpretation:

- Banks usually approve loans if **income is high, credit history is good, loan amount is reasonable compared to income, and property is in a developed area.**
- This dataset tries to simulate exactly that scenario.

## Tasks to Perform:

### 1. Data Preprocessing

- Handle missing values.
- Encode categorical features (e.g., Gender, Education).
- Apply feature scaling (especially important for Logistic Regression).
- Split the dataset into training and test sets (70/30).

### 2. Model Training (without GridSearchCV)

- Train Logistic Regression with different penalties: L1, L2, and ElasticNet.
- Train Decision Tree with basic parameters (`max_depth`, `min_samples_split`).
- Evaluate models using Accuracy, Precision, Recall, and F1 Score.

### 3. Model Training (with GridSearchCV)

- Perform hyperparameter tuning for Logistic Regression (C, penalty, solver, l1\_ratio).
- Perform hyperparameter tuning for Decision Tree (`max_depth`, `min_samples_split`, `min_samples_leaf`, criterion).
- Use **grid searchCV** to ensure balanced splits.

### 4. Comparison & Analysis

- Compare performance between Logistic Regression and Decision Tree.
- Show the effect of regularization and hyperparameter tuning.
- Discuss when Logistic Regression is preferable (interpretable, linear patterns) vs. when Decision Trees are preferable (non-linear, rule-based explanations).

## **Deliverables on Github:**

- A Jupyter Notebook with:
  - Data preprocessing steps.
  - Training and evaluation of both models (before and after GridSearchCV).
  - Performance metrics and visualizations (confusion matrix, bar plots of scores).
- A short report (1–2 pages) including:
  - Dataset description.
  - Methodology (Logistic Regression, Decision Tree, Regularization, GridSearchCV).
  - Results and analysis.
  - Conclusion: which model performed better and why.