Loan Approval Prediction – Short Report

1. Dataset Description

The dataset used in this project is the **Loan Prediction Dataset** from Kaggle, which contains information about loan applicants, including:

Column	Meaning
Loan_ID	Unique identifier for each application
Gender	Applicant's gender (Male/Female)
Married	Marital status (Yes/No)
Dependents	Number of dependents (0,1,2,3+)
Education	Education level (Graduate/Not Graduate)
Self_Employed	Employment type (Yes/No)
ApplicantIncome	Monthly income of main applicant
CoapplicantIncome	Monthly income of co-applicant
LoanAmount	Requested loan amount (in thousands)
Loan_Amount_Term	Loan term (in months)
Credit_History	Past credit repayment record (1=good, 0=bad)
Property_Area	Location type (Urban, Semiurban, Rural)
Loan_Status	Target variable: Approved (Y) / Not Approved (N)

Key points:

- Loan_ID is dropped before training.
- Categorical variables are encoded.
- Missing values are handled via mean/median imputation.
- ApplicantIncome + CoapplicantIncome combined into **TotalIncome**.
- Dataset simulates real-world bank scenarios: higher income, good credit history, reasonable loan amount, and developed property area increase approval likelihood.

2. Methodology

The goal is to predict loan approval using **Machine Learning**. Two algorithms were considered:

Logistic Regression

- Used for its **interpretability** and suitability for **linear relationships**.
- Applied **Regularization** to reduce overfitting:
    - **L1 (Lasso):** selects important features by reducing some weights to zero.
    - **L2 (Ridge):** reduces weights uniformly.
    - **Elastic Net:** combination of L1 and L2.
- **Hyperparameter tuning** via **GridSearchCV** for:
    - Penalty (L1, L2, ElasticNet, None)
    - Regularization strength (C)
    - Solver
    - L1_ratio (for ElasticNet)

Decision Tree Classifier

- Handles **non-linear patterns**.
- Hyperparameters tuned: max_depth, min_samples_split, min_samples_leaf.
- Provides intuitive **if-then rules**, but prone to overfitting.

Data Preprocessing

- Missing values handled (mean/median imputation).
- Categorical features encoded (LabelEncoder / OneHotEncoder).
- Feature scaling applied (important for Logistic Regression).
- Dataset split into **70% training / 30% testing**.

3. Results and Analysis

Model	Regularization / Hyperparameters	Train Accuracy	Test Accuracy	Notes
Logistic Regression	L2, C=0.1	0.83	0.84	Best generalization, stable weights
Logistic Regression	ElasticNet, C=0.5, l1_ratio tuned	0.85	0.84	Sparse + stable, interpretable
Logistic Regression	L1, C=1	0.84	0.81	Feature selection, slightly lower test accuracy
Decision Tree	max_depth=5	0.92	0.81	High train accuracy (overfitting), lower test accuracy

Observations:

- **Regularization and hyperparameter tuning** improved Logistic Regression performance.
- L2 and ElasticNet achieved the **highest test accuracy (0.84)**.
- Decision Tree overfits training data; test performance is lower than tuned Logistic Regression.
- Logistic Regression is preferable when interpretability and linearity assumptions hold.

4. Conclusion

- **Best performing model: Logistic Regression with L2 or ElasticNet regularization**, after hyperparameter tuning.
- **Reason:**
    - Balances **generalization vs overfitting**.
    - Provides **interpretable coefficients**, important in financial applications.
    - Performs slightly better than Decision Tree on test set.

- **Recommendation:** Use Logistic Regression for **loan approval prediction** when feature effects need to be understood and linear assumptions are reasonable.