Loan Approval Prediction

1. Dataset Description

The dataset contains information about bank loan applications. Each record includes several features that describe a customer, such as annual income, number of dependents, education level, employment type, loan amount, loan term, credit history, and different asset values. The target variable (loan_status) indicates whether the loan was approved (1) or rejected (0).

Before training, missing values were handled (numeric = median, categorical = mode), categorical columns were encoded using OneHotEncoder, and numerical features were scaled using StandardScaler. A train/test split of 70/30 was applied.

2. Methodology

2.1 Logistic Regression

Used as a baseline model because it is simple and interpretable. It works well when data relationships are linear. The model performed reasonably but did not capture the complex patterns in this dataset.

2.2 Decision Tree Classifier

A stronger baseline model that can capture nonlinear relationships. The initial model performed well but had a risk of overfitting. Hyperparameter tuning was later used to improve generalization.

2.3 Regularization

Regularization was mainly applied through Logistic Regression using the parameter C, which controls penalty strength. The effect was minimal because the model itself was not the best architecture for this dataset.

2.4 GridSearchCV

Used to tune hyperparameters for the following models: Logistic Regression, Decision Tree, Random Forest, and AdaBoost.

GridSearchCV systematically searches for the best combination of hyperparameters and helps reduce overfitting while improving test performance.

3. Results and Analysis

A comparison table was created after training all models (baseline and tuned). Key observations:

• Logistic Regression: Stable but lowest performance (~91–92% accuracy).

• Decision Tree: Strong performance; improved after tuning (≈ 98.4% accuracy).

• Random Forest: Very high accuracy (~98%), slight overfitting.

• AdaBoost: Improved significantly after tuning (≈ 97.9% accuracy).

• Gradient Boosting: Best overall performance with ≈ 98.67% accuracy.

• XGBoost: Also strong (≈ 98.4%) but showed overfitting on the training set.

Gradient Boosting had the most balanced results (train ≈ test), showing strong generalization without overfitting.

4. Conclusion

Among all tested models, Gradient Boosting performed the best.

Reasons:

• Highest test accuracy (≈ 98.67%)

• Minimal difference between train and test accuracy

• Excellent handling of nonlinear relationships

• More stable results compared to Random Forest and XGBoost

Decision Tree (tuned) and XGBoost were close competitors, but Gradient Boosting offered the best balance between accuracy and generalization.