***Loan Eligibility Prediction Project Report***

***1. Introduction***

**Objective**: Predict loan eligibility using applicant data to automate and optimize the approval process.

**Algorithm**: Base Learner: Random Forest Classifier, Support Vector Classifier, XGBoost Classifier

Meta Learner: Logistic Regression Algorithm.

**Dataset:** [Loan Prediction Problem Dataset](https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset)

***2. Data Loading & Initial Exploration***:

* **Source**: train\_ctrUa4K.csv
* **Features**:

Loan\_ID: Unique identifier

Categorical: Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Amount\_Term, Credit\_History, Loan\_Status: Target.

Numerical: ApplicantIncome, CoapplicantIncome, LoanAmount.

* **Initial Inspection**:

count mean std min 25% 50% 75% max

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ApplicantIncome** |  | 614.0 | 5403.0 | 6109.0 | 150.0 | 2878.0 | 3812.0 | 5795.0 | 81000.0 |
| **CoapplicantIncome** |  | 614.0 | 1621.0 | 2926.0 | 0.0 | 0.0 | 1188.0 | 2297.0 | 41667.0 |
| **LoanAmount** |  | 592.0 | 146.0 | 86.0 | 9.0 | 100.0 | 128.0 | 168.0 | 700.0 |
| **Loan\_Amount\_Term** |  | 600.0 | 342.0 | 65.0 | 12.0 | 360.0 | 360.0 | 360.0 | 480.0 |
| **Credit\_History** |  | 564.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 |

* **No Duplicates**: Confirmed via df.duplicated().sum().
* **Missing Values**: Gender (13), Married (3), Dependents (15), Self\_Employed (32), LoanAmount (22), Loan\_Amount\_Term (14), Credit\_History (50).

3. ***Exploratory Data Analysis (EDA)***

**Handling Missing Values**:

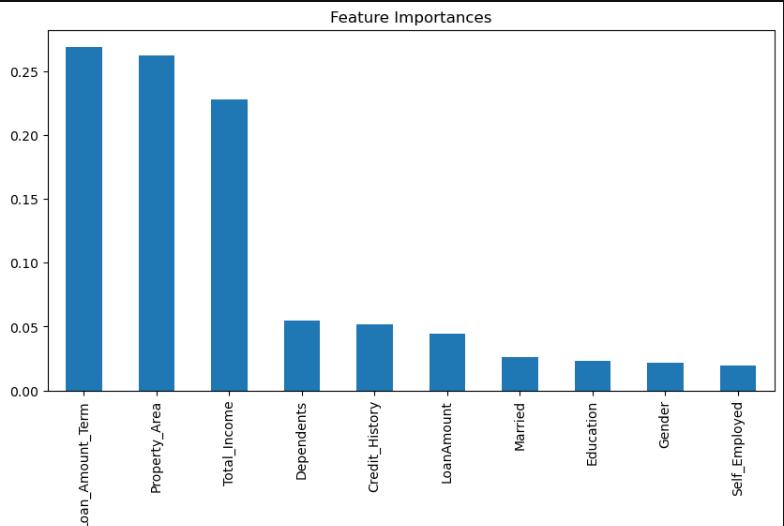
* Gender: Replaced NaN with mode ("Male").
* Married: Replaced NaN with mode ("Yes").
* Dependents: Replaced NaN with mode ("0").
* Self\_Employed: Replaced NaN with mode ("No").
* LoanAmount: Imputed using median grouped by Education.
* Loan\_Amount\_Term: Replaced NaN with mode (360 days).
* Credit\_History: Replaced NaN with mode (1.0).

**Categorical Feature Encoding**:

|  |  |
| --- | --- |
| *Feature name* | *Encoding* |
| Gender | Male: 1, Female: 0 |
| Married | Yes: 1, No: 0 |
| Education | Graduate: 1, Not Graduate: 0 |
| Self\_Employed | Yes: 1, No: 0 |
| Property\_Area | Rural: 0, Urban: 1, Semiurban: 2 |
| Loan\_Status | Y: 1, N: 0 |
| Dependents | Ordinal (0, 1, 2, 3) |

**Feature Engineering**:

* Created Total\_Income: Sum of ApplicantIncome and CoapplicantIncome. Now dropped ApplicantIncome and CoapplicantIncome columns.
* With the help of Ensemble Learning Technique (Random Forest Algorithm) we can find independent feature is more important to predict the target variable to reduce model complexity and prevent to overfitting the model. The importance of feature are ranking in following image:



**Dataset After Preprocessing**:

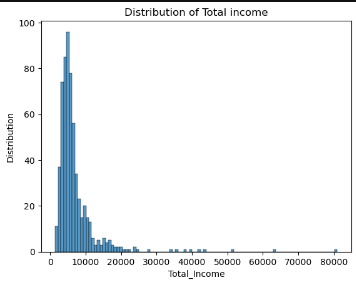
Features: 12 (excluding Loan\_ID).

Missing Values: 0.

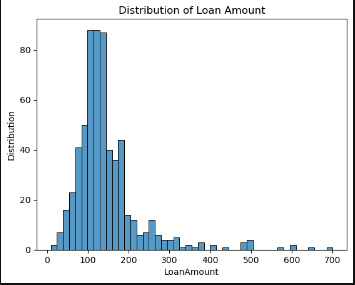
Sample Size: 614 entries.

4. ***Data Visualization & Insights***

* **Distribution of Total Income**:



* **Distribution of Loan Amount:**

****

* **Target Imbalance**:

Loan\_Status: 422 approved (Y), 192 rejected (N).

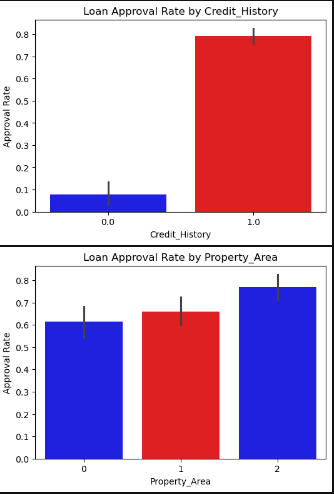
* **Key Influencers**:

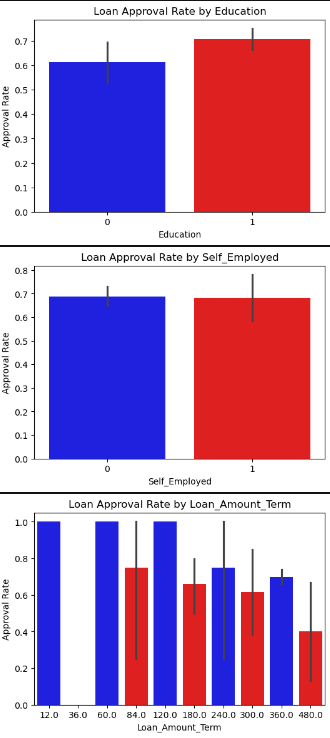
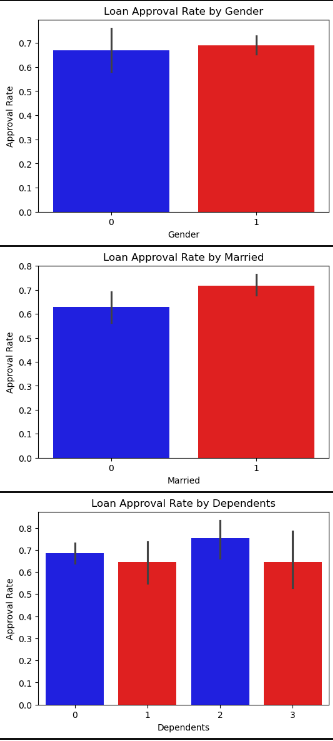
Credit\_History: Crucial (475 with "1" vs. 89 with "0").

Property\_Area: Semiurban (233) > Urban (202) > Rural (179).

* **Other Trends**:

1. Married applicants (398) more common than unmarried (213).
2. Graduates (480) dominate non graduates (134).
3. Loan Approval Rate by {Gender, Married, Dependents, Education, Self Employed, Property Area, Loan Amount Term, Credit History}

34).

h

***5. Model Building: Random Forest, Support Vector Classifier, XGBoost and Logistic Regression*:**

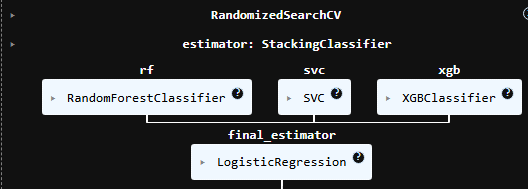
* **Data Splitting**:
* **Features**: Married, Dependents, LoanAmount, Loan\_Amount\_Term, Credit\_History, Property\_Area, Total\_Income.
* **Target**: Loan\_Status.
* **Split Ratio**: 70% training, 30% testing.
* **Model Training**: Using Ensemble Technique (Stacking Classifier)

Why use Stacking Ensemble Technique?

* **Stacking** is an **ensemble learning technique** that combines multiple classification or regression models (called **base learners**) to get a better predictive performance than any single model alone.

Base learner:  ***Random Forest, Support Vector Classifier, XGBoost***.

Meta Learner: ***Logistic Regression***.



***Hyperparameter Tunning Using RandomizedSearchCV***

*RandomizedSearchCV* is a technique used for *hyperparameter tuning* in machine learning. Instead of trying *every possible combination* of hyperparameters like GridSearchCV does, RandomizedSearchCV:

* **Randomly samples** combinations from the hyperparameter space.
* **Trains and evaluates** the model for a fixed number of combinations (n\_iter).

**Why do we use RandomizedSearchCV?**

**1. Faster than Grid Search**

* Grid Search checks all combinations, which can be very slow.
* Randomized Search checks only a random subset, saving time.

**2. Good for Large Search Spaces**

* If you have many hyperparameters or large ranges (e.g., C from 1 to 10,000), GridSearch becomes infeasible.
* RandomizedSearch can still find near-optimal values quickly.

Print(random\_search\_stack.best\_params\_):

{**'final\_estimator\_\_C': 1,**

**'estimators': [('rf',**

**RandomForestClassifier(max\_depth=5, n\_estimators=150, random\_state=42)),**

**('svc', SVC(class\_weight='balanced', probability=True, random\_state=42)),**

**('xgb',**

**XGBClassifier(base\_score=None, booster=None, callbacks=None,**

**colsample\_bylevel=None, colsample\_bynode=None,**

**colsample\_bytree=None, device=None, early\_stopping\_rounds=None,**

**enable\_categorical=False, eval\_metric='logloss',**

**feature\_types=None, feature\_weights=None, gamma=None,**

**grow\_policy=None, importance\_type=None,**

**interaction\_constraints=None, learning\_rate=None, max\_bin=None,**

**max\_cat\_threshold=None, max\_cat\_to\_onehot=None,**

**max\_delta\_step=None, max\_depth=None, max\_leaves=None,**

**min\_child\_weight=None, missing=nan, monotone\_constraints=None,**

**multi\_strategy=None, n\_estimators=None, n\_jobs=None,** **random\_search\_stack.best\_params\_**

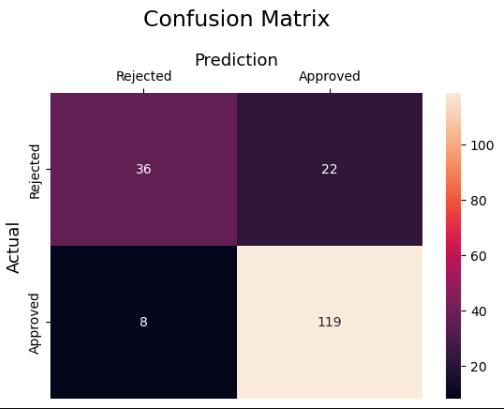
**num\_parallel\_tree=None))]}**

***6. Model Evaluation:***

* **Accuracy**: ~85%
* **Precision/Recall**: Class 0 (Rejected): Precision ~0.75, Recall ~0.60.

Class 1 (Approved): Precision ~0.88, Recall ~0.93.

* **Confusion Matrix**:



* **ROC-AUC Score**: ~78%

***7. Conclusion***

After identifying the top 7 important features using Random Forest, a stacking ensemble model was built using:

Random Forest

Support Vector Machine (SVC)

XGBoost

...with Logistic Regression as the meta-classifier.

The model was tuned using RandomizedSearchCV for optimal performance

***Project Impact***:

Automates loan eligibility assessment, reduces processing time, and minimizes human bias.