A Case Study Report on

**Loan Approval Prediction**

by

# Vansh Jeetendra Jain, 20102156

# Ghanshyam Kachhia, 20102026

in the subject

# Applied Data Science



Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

G.B. Road, Kasarvadavli, Thane(W)

University of Mumbai

2023-2024

## 1. Problem Statement:

Large volumes of loan applications are manually reviewed by financial organisations, which can cause irregularities and long processing periods. The goal of this project is to create a machine learning model that uses applicant data to forecast loan approvals. The model will assess variables such as loan amount, credit history, income, and demographics in order to calculate the probability of a successful repayment. This will facilitate data-driven lending decisions, increase efficiency, and streamline the loan approval procedure. The model's ability to identify creditworthy borrowers and predict loan approvals accurately will be used to gauge the project's performance.

**2. Description of dataset:**

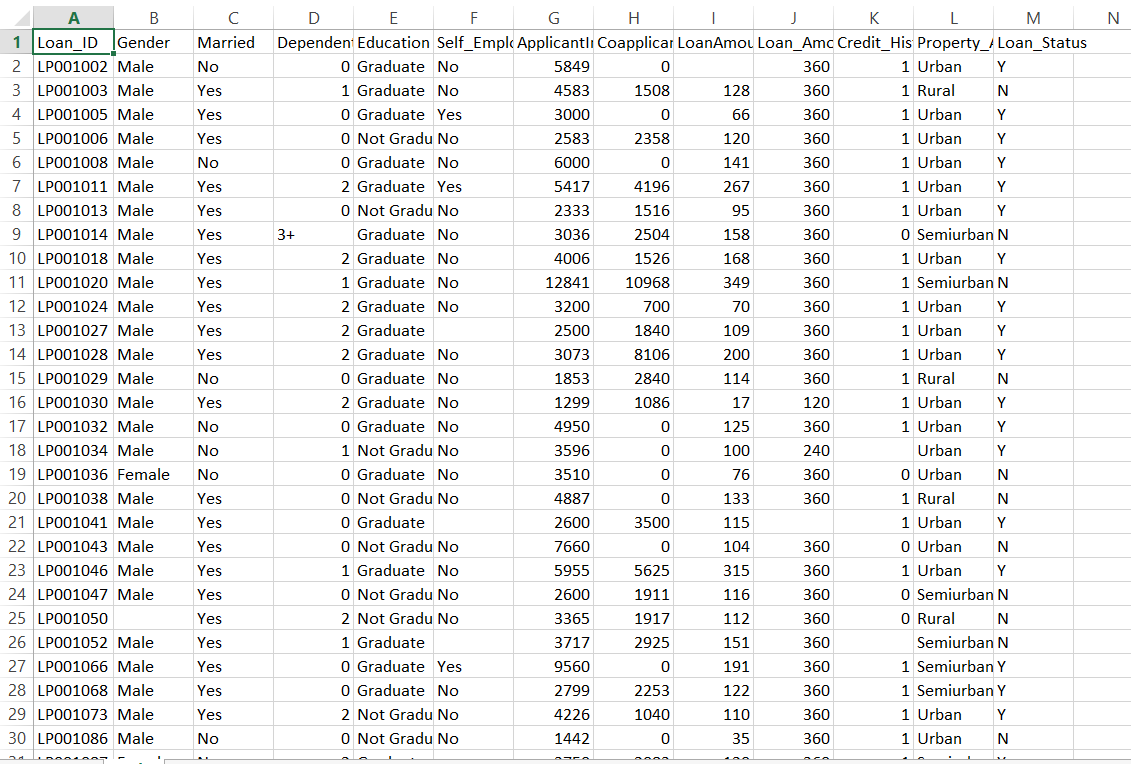


Fig. 1 Loan Application Dataset.

**Size:** The dataset comprises a single CSV file, typically with multiple(614) rows (instances) and 13 columns (features).

**Features:** It includes various features that are potentially useful for predicting loan approval decisions. Some common features in such datasets include:

**Applicant ID:** Unique identifier for each loan applicant.

**Gender:** Gender of the applicant (e.g., Male, Female).

**Marital Status:** Marital status of the applicant (e.g., Married, Single, Divorced).

Education: Educational qualification of the applicant (e.g., Graduate, Not Graduate).

**Dependents**: Number of dependents of the applicant.

**Applicant Income:** Income of the applicant.

Coapplicant Income: Income of the co-applicant (if any).

**Loan Amount:** Amount of the loan requested by the applicant.

**Loan Term:** Term of the loan (e.g., 15 years, 30 years).

**Credit History:** Credit history of the applicant (e.g., 1 for good credit history, 0 for poor credit history).

**Property Area:** Area where the property is located (e.g., Urban, Semiurban, Rural).

Loan Status: The target variable indicating whether the loan was approved or not (e.g., Approved, Not Approved).

**Objective:** The primary objective of using this dataset is to develop a predictive model that can accurately predict whether a loan application will be approved or denied based on the provided features. This prediction can help financial institutions streamline their loan approval process and make more informed decisions.

3. **Descriptive statistics and inferential statistics:**

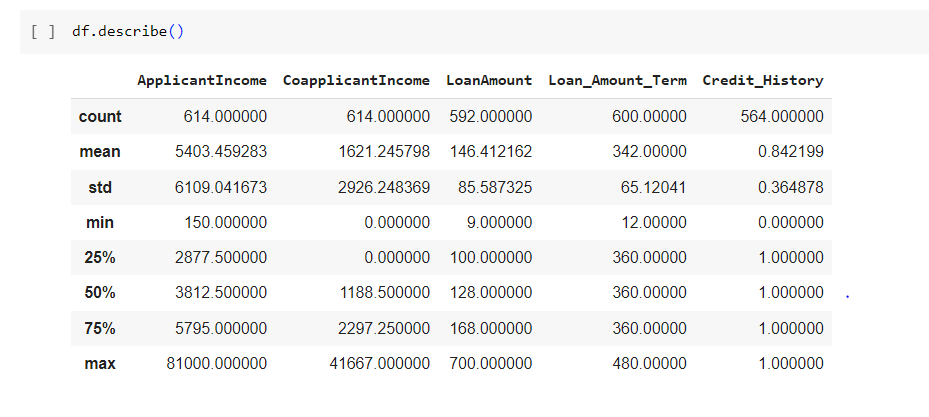


Fig. 2 Descriptive Statistics.

The image is a table containing summary statistics for several numerical variables.

The first row of the table shows the names of the variables. These include ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, and Credit\_History.

The second row shows the number of non-missing values (count) for each variable. For example, there are 614 non-missing values for ApplicantIncome, but only 592 for LoanAmount.

The third row shows the mean (average) of each variable. For example, the average applicant income is $5403.46.

The fourth row shows the standard deviation (spread) of each variable. For example, there is a standard deviation of $6109.04 in applicant income.

The fifth row shows the minimum value of each variable. For example, the minimum applicant income is $150.

The sixth row shows the 25th percentile (first quartile) for each variable. For example, the 25th percentile of applicant income is $2877.50. This means that 25% of applicants have an income below $2877.50.

The seventh row shows the 50th percentile (median) for each variable. For example, the median applicant income is $3812.50. This means that 50% of applicants have an income below $3812.50 and 50% have an income above $3812.50.

The eighth row shows the 75th percentile (third quartile) for each variable. For example, the 75th percentile of applicant income is $5795.00. This means that 75% of applicants have an income below $5795.00.

The ninth row shows the maximum value of each variable. For example, the maximum applicant income is $81000.00.

Overall, the table provides a summary of the distribution of these financial measures in a dataset of loan applications.

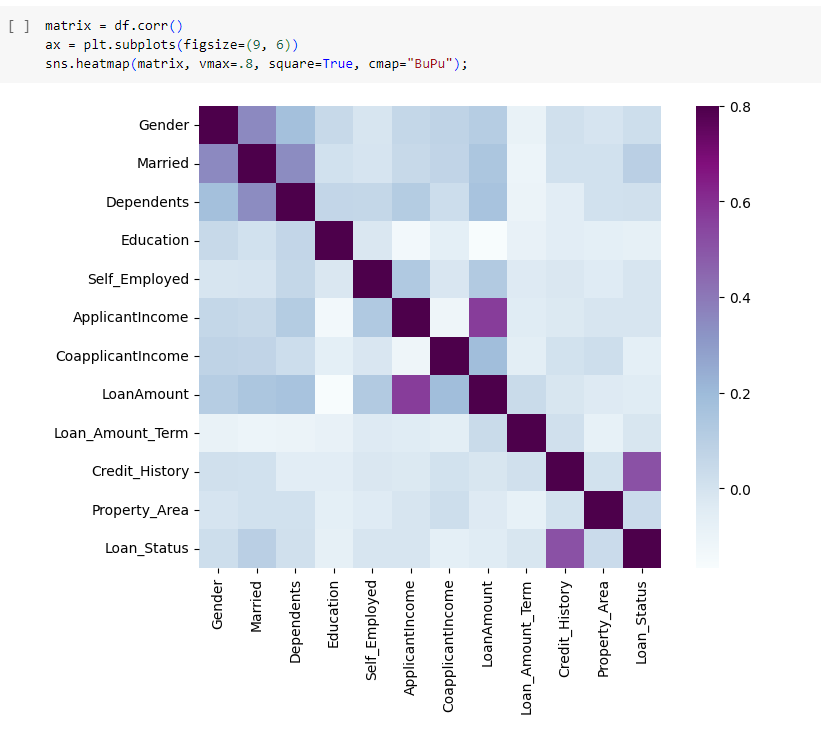


Fig. 3 Inferential Statistics.

The image is a heatmap, which is a graphical representation of a correlation matrix. A correlation matrix is a table that shows the correlation coefficient between all pairs of features in a dataset. The correlation coefficient is a measure of the linear dependence between two variables. It ranges from -1 to 1, with -1 indicating a perfect negative correlation, 0 indicating no correlation, and 1 indicating a perfect positive correlation.

In the heatmap, the color intensity of each cell represents the correlation coefficient between the two features corresponding to the row and column labels of that cell. For instance, the dark blue cell in the upper left corner of the heatmap indicates a strong positive correlation between Gender and Married.

Here’s a description of the correlations represented in the heatmap:

* Gender and Married have a strong positive correlation. This means that people who are identified as one gender are more likely to be married.
* Gender and Dependents have a moderate positive correlation. This means there is a weak tendency for people of a particular gender to have more dependents.
* There is a weak positive correlation between Married and Dependents, and Education and Dependents. This means that married couples and more educated people tend to have more dependents.
* Self\_Employed has a weak positive correlation with ApplicantIncome and CoapplicantIncome. This means that self-employed people tend to have a slightly higher income than non-self-employed people. There is also a weak positive correlation between ApplicantIncome and CoapplicantIncome, which means that people with higher incomes tend to have co-applicants with higher incomes.
* LoanAmount has a weak positive correlation with ApplicantIncome, CoapplicantIncome, and Loan\_Amount\_Term. This means that people with higher incomes and co-applicants with higher incomes tend to apply for larger loans for longer terms.
* Credit\_History has a weak positive correlation with LoanAmount. This means that people with better credit histories tend to apply for larger loans.
* There is no correlation between Property\_Area and Loan\_Status. This means that where someone lives has no bearing on whether their loan is approved.

**4. Data Preprocessing:**

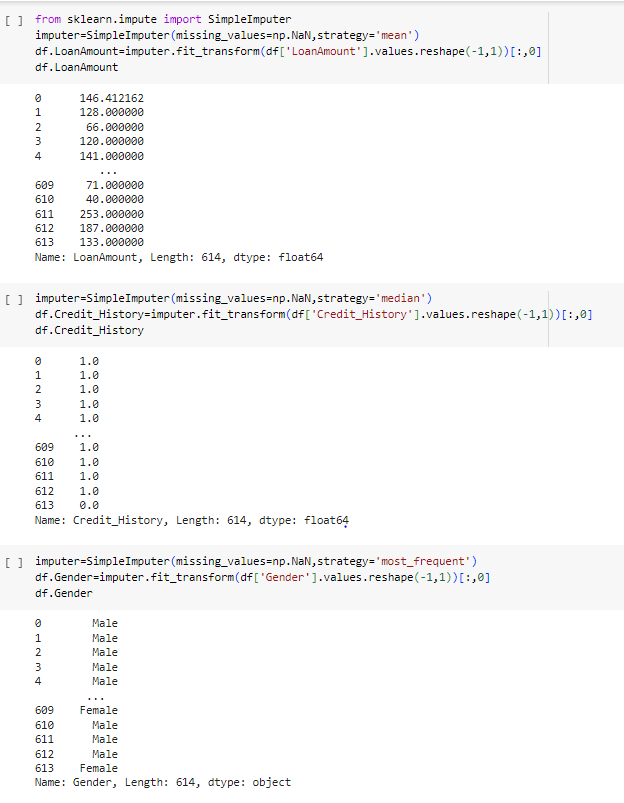


Fig. 4 Handling Missing Values.

The image is a Python code that addresses missing values in a financial dataset, likely loan applications. It imports tools from scikit-learn to impute missing values in specific columns ("LoanAmount", "Credit\_History", "Gender") using different strategies like replacing with the mean, median or most frequent value depending on the column. The code then displays the resulting DataFrame with imputed values.

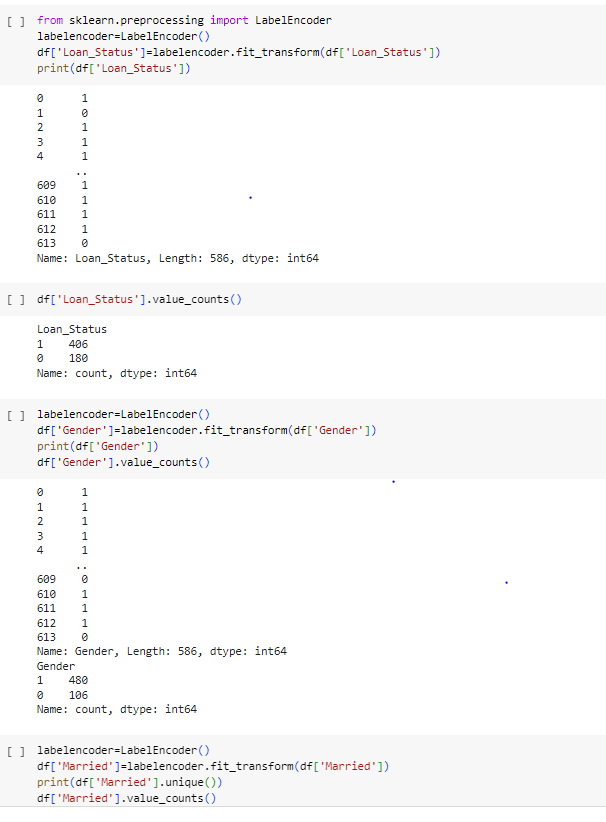


Fig. 5 Label Encoding.

The provided Python code snippet demonstrates label encoding, a technique to convert categorical data into numerical data suitable for machine learning models. It imports the LabelEncoder class from scikit-learn, creates an instance, and fits it to a DataFrame's column (e.g., Loan\_Status) to assign unique integers to each category. This process is repeated for other categorical columns, transforming the data for further analysis.

## 5. Data Visualization:

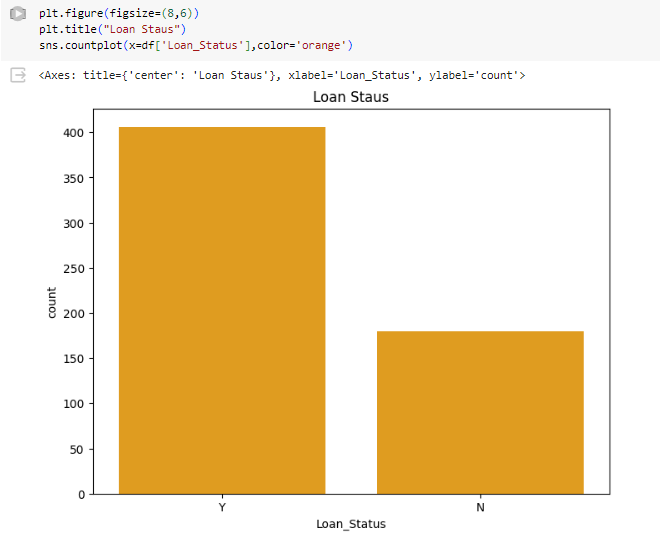


Fig. 6 Visualization of approved and rejected loans.

The image is a countplot which shows the number of loans that were approved and rejected. In this case, it appears that more loans were approved (Y) than rejected (N).

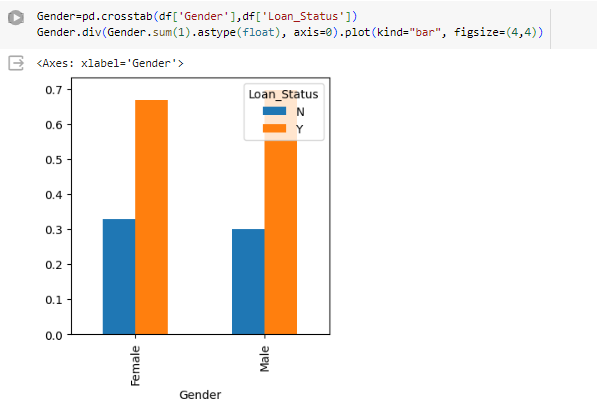


Fig. 7 Loan Approval Rates by Gender

The image you sent is a bar graph that shows the percentage of people who have a loan by gender. The y-axis shows the percentage and the x-axis shows gender. It appears that a higher percentage of females have loans than males.

## 6. Model Building:

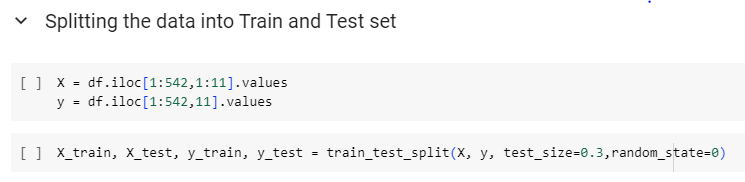


Fig. 8 Splitting Data into Training and Testing Sets

This Python code splits data into training and testing sets. It extracts features (X) and target variables (y) from a DataFrame and uses train\_test\_split to divide them. The test\_size parameter allocates 30% of the data for testing, while the remaining 70% goes to training. Setting random\_state ensures consistent data splits if the code is run multiple times.

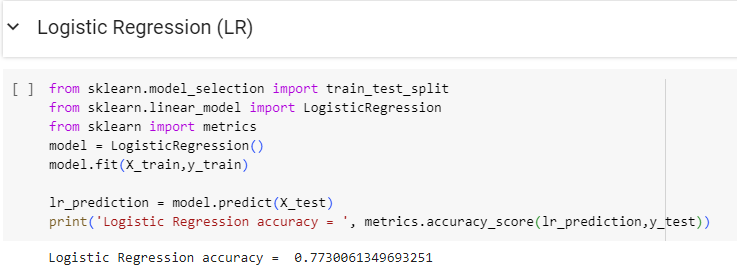


Fig. 9 Logistic Regression Model

This Python code implements logistic regression using scikit-learn. It first imports the LogisticRegression class and creates a model object. Then, it trains the model on the provided features (X\_train) and target variables (y\_train). The trained model can then predict the target variable for new data (X\_test).

## 7. Model Validation:

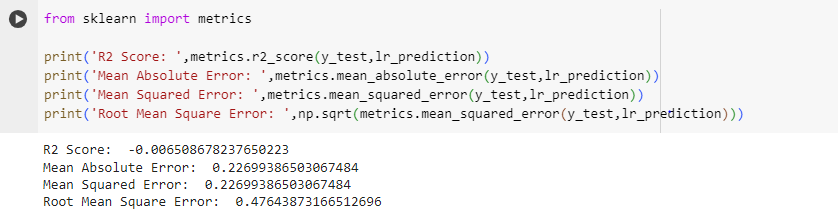


Fig. 10 Evaluating Linear Regression Model Performance

This Python code calculates performance metrics for a linear regression model. It imports functions from scikit-learn to compute the R-squared score (goodness of fit), mean squared error (average squared difference between predictions and actual values), and root mean squared error (square root of MSE, in the same units as predictions) for the model's predictions (lr\_prediction) on the test data (y\_test). These metrics help assess how well the model generalizes to unseen data.