

Automating bank credit processes using AI

Mindverse

Introduction

The aim of this project is to build a platform that can help banks in deciding whether or not to give someone a loan. This project can help banks reduce their risk as well as make them faster and more efficient in processing loan requests.

Project structure

We will provide a Drive link which includes all of the following documents:

```
/project
├── Automating_bank_credit_processes_using_AI.pdf
├── loan_data_set.csv
├── challenge-proxym.ipynb
├── model.txt
├── Platform/
│   ├── back_end/
│   └── front_end/
```

Data

We used a Dataset from Kaggle which can be found here: <https://www.kaggle.com/datasets/burak3ergun/loan-data-set>

The data set has 614 rows and 13 variables including 8 categorical variables, 4 continuous variables, and 1 variable to accommodate the loan ID.

This is a description of each variable:

- Loan_ID: Loan reference number
- Gender: Applicant gender
- Married: Applicant marital status
- Dependents: Number of family members
- Education: Applicant education/qualification
- Self_Employed: Applicant employment status

- ApplicantIncome: Applicant's monthly salary/income
- CoapplicantIncome: Additional applicant's monthly salary/income
- LoanAmount: Loan amount
- Loan_Amount_Term: The loan's repayment period (in days)
- Credit_History: Records of previous credit history (0: bad credit history, 1: good credit history)
- Property_Area: The location of property (Rural/Semiurban/Urban)
- Loan_Status Status of loan (Y: accepted, N: not accepted)

Preliminary study, training & evaluation

In the challenge-proxym.ipynb file we start by exploring the data: understanding each feature and how they relate to our target. We then process the data and train different models to predict our goal. We finally evaluate each model to choose the best one. The results of this evaluation is described in the Table 1

Algorithm	accuracy
Catboost	76.42%
Xgboost	74.79%
LightGBM	77.23%
Voting system	75.60%

Table 1: Comparing the accuracy of different systems

Thus we chose the LightGBM model which is a good choice not only for its accuracy but another benefit to using it is that it can handle missing data which is going to be helpful in the platform.

Platform

We also created a platform from which you can use the model. You can input as much data as you want then press "Analysis" and you get the model's decision as shown in Figure 1. The back-end is built using Python and Flask and the front-end is built using JavaScript and VueJS. To run this project you will have to run the back-end first. You can install the necessary libraries using the requirements file. You can then run the back-end server by running the script main.py. Then you can access the front end by opening the index.html file in your browser and using it.

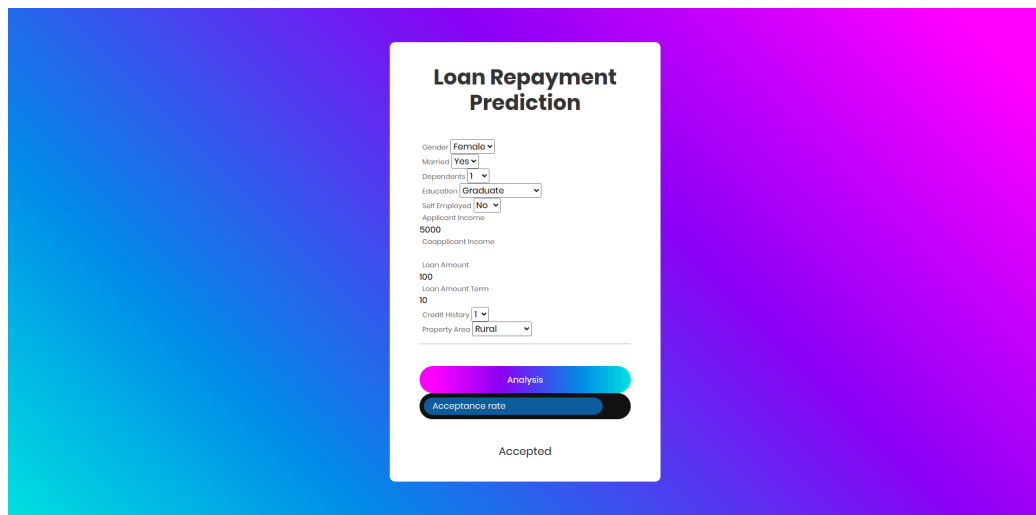
The image shows a web application interface for "Loan Repayment Prediction". The interface is centered on a white card against a background with a blue-to-purple gradient. The card contains several input fields: "Gender" (dropdown menu showing "Female"), "Married" (dropdown menu showing "Yes"), "Dependents" (dropdown menu showing "1"), "Education" (dropdown menu showing "Graduate"), "Self Employed" (dropdown menu showing "No"), "Applicant Income" (text input showing "5000"), "Loan Amount" (text input showing "100"), "Loan Amount Term" (text input showing "10"), "Credit History" (dropdown menu showing "1"), and "Property Area" (dropdown menu showing "Rural"). Below these inputs are two buttons: a blue "Analysis" button and a black "Acceptance rate" button. At the bottom of the card, the word "Accepted" is displayed.

Figure 1: The platform interface