

## Loan Prediction Challenge

The goal is to have a web app where a user provides data for a loan applicant and finds out whether they are eligible for loan. The app should consist of a pre-trained classifier, a frontend, and a backend.

### Dataset

Find attached “train.csv” and the data description below.

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant Married (Y/N)
Dependents	Number of Dependents
Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self Employed (Y/N)
ApplicantIncome	Applicant Income
CoapplicantIncome	Co-applicant Income
LoanAmount	Loan Amount in Thousands
Loan_Amount_Term	Term of Loan in Months
Credit_History	Credit History meets Guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	(Target) Loan approved (Y/N)

## Components of the Task

### Task 1: - Data analysis and ML Classifier

- State your hypothesis
- Show your work on data exploration, variable analysis & data pre-processing in a separate Jupyter notebook
- Show your work on modelling with at least 2 trained ML models using the given data for prediction
- Use the best among all models for predicting the outcome of the loan application
- Submit best pre-trained model for predicting loan application outcome integrated into the web app

### Task 2.1: - Frontend

- Allow the user to input the loan application parameters
- Show output: “Eligible” or “Not Eligible” based on the response of the request to the backend
- Note: A simple UI that can interact successfully with the backend and can take responses from it will be acceptable

### Task 2.2: - Backend

- Create REST API to determine the result of the loan application
- Use the pre-trained model to determine the outcome of the loan application

## Submission

- Submit your work on private Git repo and share the link of repo with Feige team
- Use Jupyter notebook to showcase your work
- Share the final trained model file with prediction pipeline code to make a prediction on test data which we use for the trained model prediction separately
- Share the properly executed webapp from frontend to backend. There is no need to make it in Jupyter notebook, it can be a separate python app with instructions on how to bring it up on new machine
- **Note-1:** Feel free to use your creativity to show your work
- **Note-2:** Train the model with different input variable combinations and show your work if you find interesting input combination for better predictions
- **Note-3:** Include observations on business implications for outcomes if possible