CREDIT CARD DEFAULT PREDICTION

Low Level Design (LLD)

Manisha Singh, August 2023

Document Version Control

Date Issued	Version	Description	Author
August 2, 2023	1.0	HLD	Manisha Singh

Contents

Abstract

The project "Credit Card Default Prediction" aims to develop an accurate and efficient prediction model to identify customers at risk of defaulting on their credit card payments. Defaults on credit card payments are a significant issue for financial institutions, leading to significant losses and potential economic instability.

To address this challenge, the project deploys advanced machine learning algorithms and techniques to analyse historical transaction data, customer behaviour, and socio-economic factors. The dataset contains various features such as payment history, credit utilization, age, income, and other relevant variables.

1. Introduction

1.1. Why this Low-Level Design Document?

The goal of LLD or a Low-Level Design document is to give the internal logical design of the actual program code. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Technical Specifications

2.1. Dataset

File Name	Finalized	Source
UCI_Credit_Card.csv	Yes	https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

2.1.1. Dataset Overview

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

2.1.2. Input Schema

Feature Name	Datatype	Null/Required
ID	Integer	Required
LIMIT_BAL	Integer	Required
SEX	Integer	Required
EDUCATION	Integer	Required
MARRIAGE	Integer	Required
AGE	Integer	Required
PAY_0	Integer	Required
PAY_2	Integer	Required
PAY_3	Integer	Required
PAY_4	Integer	Required
PAY_5	Integer	Required
PAY_6	Integer	Required
BILL_AMT1	Integer	Required
BILL_AMT2	Integer	Required
BILL_AMT3	Integer	Required
BILL_AMT4	Integer	Required
BILL_AMT5	Integer	Required
BILL_AMT6	Integer	Required
PAY_AMT1	Integer	Required
PAY_AMT2	Integer	Required
PAY_AMT3	Integer	Required
PAY_AMT4	Integer	Required
PAY_AMT5	Integer	Required
PAY_AMT6	Integer	Required
default.payement.next.month	Integer	Required

2.2. Predicting Credit Fault

- The system presents the set of inputs from the user.

• The user gives required information.

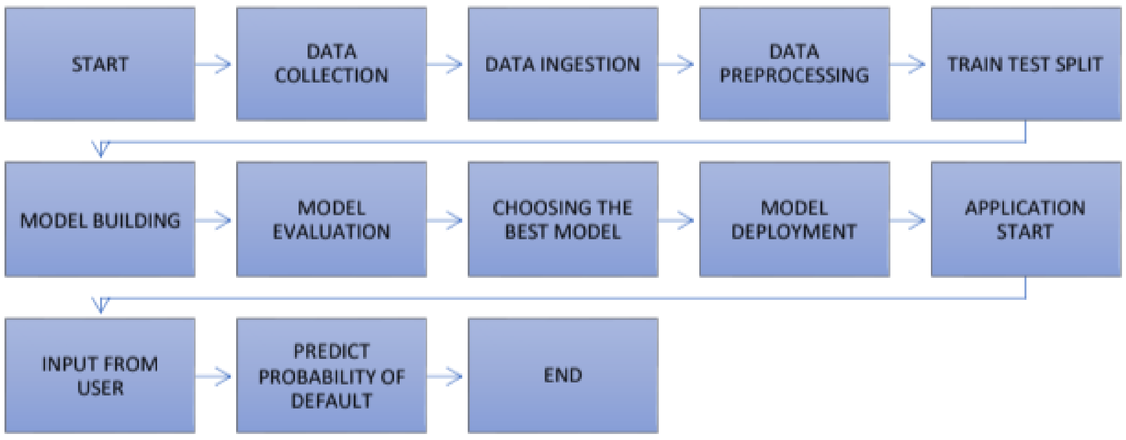
• The system should be able to predict whether the customer is likely to default in the following month.

2.3. Logging

We should be able to log every activity done by the user.

- The system identifies at what step the logging is required.
- The system should be able to log every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.

3. Architecture



4. Performance

4.1. Data Description

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

There are 25 variables:

- ID: ID of each client
- LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)
- AGE: Age in years
- PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
- PAY_2: Repayment status in August, 2005 (scale same as above)
- PAY_3: Repayment status in July, 2005 (scale same as above)
- PAY_4: Repayment status in June, 2005 (scale same as above)
- PAY_5: Repayment status in May, 2005 (scale same as above)
- PAY_6: Repayment status in April, 2005 (scale same as above)
- BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
- BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
- BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
- BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
- BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
- BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
- PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
- PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
- PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
- PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
- PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
- PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
- default.payment.next.month: Default payment (1=yes, 0=no)

4.2. Data Exploration

We divide the data into two types: numerical and categorical. We explore through each type one by one. Within each type, we explore, visualize and analyze each variable one by one and note down our observations. We also make some minor changes in the data like change column names for convenience in understanding.

4.3. Feature Engineering

Encoded categorical variables.

4.4. Train/Test Split

Split the data into 70% train set and 30% test set.

4.5. Model Building

Built models, trained, and tested the data on the models. Compared the performance of each model and selected the best one.

4.6. Save the Model

Saved the model by converting into a pickle file.

4.7. Application Start and Input Data by the User

Start the application and enter the inputs.

4.8. Prediction

After the inputs are submitted the application runs the model and makes predictions. The output is displayed as a message indicating whether the customer whose demographic and behavioral data are entered as inputs, is likely to default in the following month or not.

5. Unit Test Cases

Test Case Description	Pre-requisite	Expected Result
Verify whether the Application URL is accessible to the user.	1. Application URL should be defined.	Application URL should be accessible to the user.
Verify whether the Application loads completely for the user when the URL is accessed.	1. Application URL is accessible. 2. Application is deployed.	The Application should load completely for the user when the URL is accessed.
Verify whether user is able to see input fields on logging in.	1. Application URL is accessible. 2. Application is deployed.	User should be able to see input fields on logging in.
Verify whether user is able to edit all input fields.	1. Application URL is accessible. 2. Application is deployed.	User should be able to edit all input fields.
Verify whether user gets Submit button to submit the inputs.	1. Application URL is accessible. 2. Application is deployed.	User should get Submit button to submit the inputs.
Verify whether user is presented with recommended results on clicking submit.	1. Application URL is accessible. 2. Application is deployed.	User should be presented with recommended results on clicking submit.