CREDIT CARD DEFAULT PREDICTION

Low Level Design Document

Prepared By:
Binaya Kumar Pradhan

Abstract

Credit risk plays a significant role in the banking industry's operations, which encompass services like providing loans, credit cards, investments, mortgages, and various financial offerings. Among these, credit card services have witnessed substantial growth in recent years. However, the increasing number of credit card users has presented a challenge for banks in the form of a rising default rate. In response to this issue, data analytics can offer viable solutions for managing credit risks effectively.
This project focuses on the development of a predictive model designed to assess whether a given credit card holder is likely to default in the upcoming month. The model utilizes demographic and behavioral data from the preceding six months to make these predictions.

Table of Contents

1. Introduction

1.1.What is LLD?

The primary objective of this document is to provide an in-depth description of the Deep EHR System. It aims to elucidate the system's objectives and functionalities, outline its interfaces, define its core functions, address operational constraints, and detail its responses to external inputs. This document serves as a valuable resource for both stakeholders and developers involved in the system's development and is intended for submission to senior management for their approval.

1.2.Scope

The Deep EHR System will be a web-based application designed to forecast the likelihood of customers defaulting on credit payments at the earliest possible stage. This predictive tool leverages historical Electronic Health Records (EHR) data to enhance disease management and intervention strategies. The system's primary purpose is to predict credit card defaults based on customer information, including demographic data and credit payment history.

2. Technical Specification

2.1.Dataset

Source URL: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

2.1.1. Dataset Overview

The dataset comprises a single table, labelled "UCI_Credit_Card," which includes both personal information and historical payment data for approximately 30,000 customers. This payment data covers the preceding six months, from April to September.

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_1	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. payment. next. Month	Integer	Required

2.2.Predicting Credit Fault

- The system initiates by presenting a user interface with input fields.
- The user then provides the necessary information.
- Subsequently, the system is tasked with making a prediction regarding the customer's likelihood of defaulting in the upcoming month.

2.3.Logging

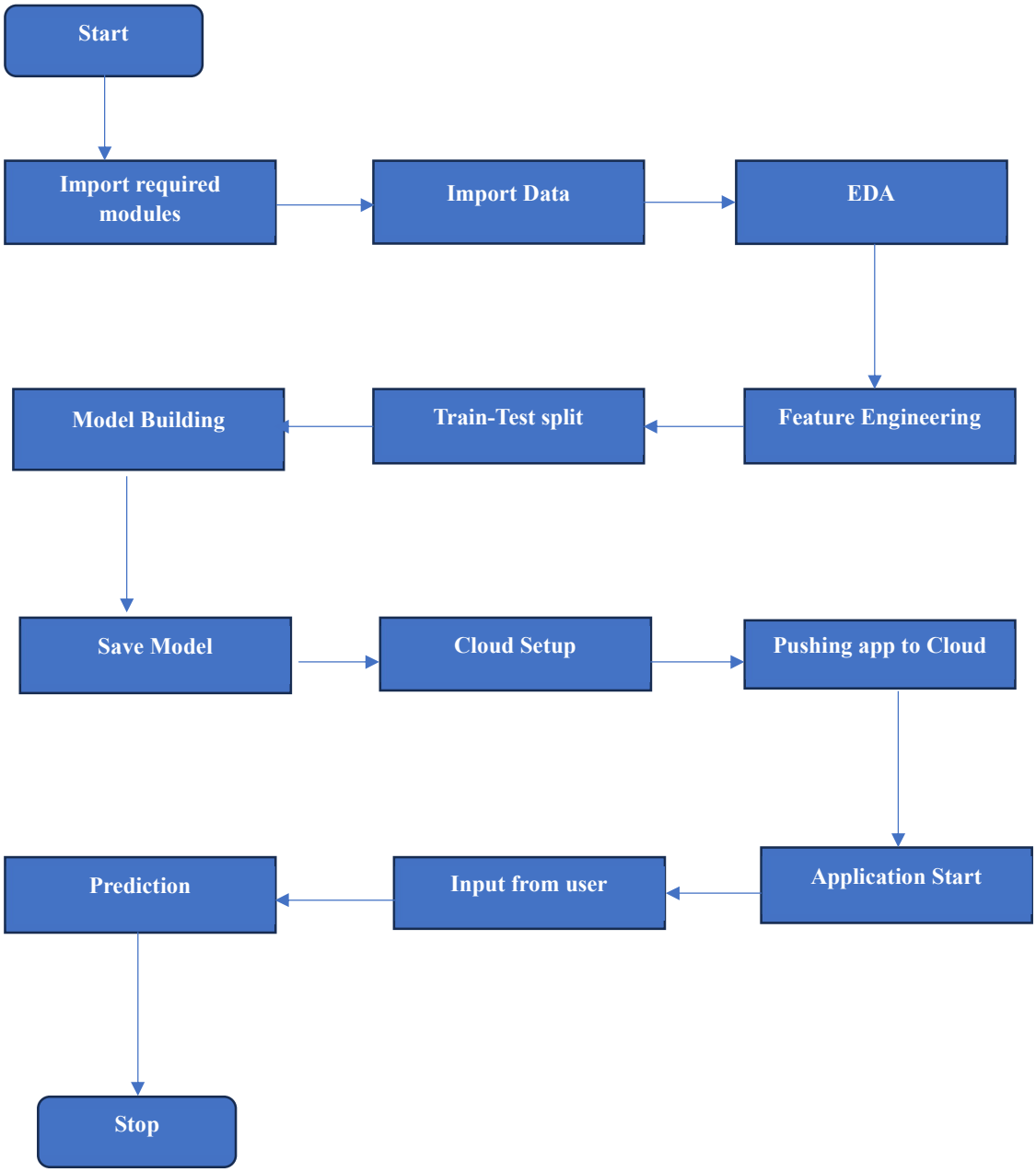
It's essential to maintain a comprehensive log of every user activity. The system autonomously recognizes the appropriate moments for logging.

- Logging is obligatory for all system processes and flows.
- Developers have the flexibility to select their preferred logging methods, whether it's database logging or file logging.
- Importantly, the system must remain operational and not experience performance issues, even with extensive logging. This emphasis on logging is primarily for effective issue debugging and, thus, is a mandatory practice.

2.4.Deployment

Deployed in AWS Elastic Beanstalk

3. Architecture



4. Architecture Description

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,
  - 0, 4, 5, 6=others)
- MARRIAGE: Marital status
  - 1=married,
  - 2=single,
  - 3=divorce,
  - 0=others
- AGE: Age in years
- PAY_0: Repayment status in September, 2005
  - -2: No consumption;
  - -1: Paid in full;
  - 0: The use of revolving credit;
  - 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

The data is categorized into two types: numerical and categorical. We conduct a detailed exploration for each type, one at a time. Within each type, we systematically examine, visualize, and analyze each variable individually, documenting our findings. Additionally, we may make minor modifications to the data, such as renaming columns for improved clarity and ease of understanding.

4.3. Feature Engineering

Categorical variables have been encoded to facilitate data analysis and modelling

4.4. Train Test Split

The dataset has been divided into two subsets: a training set, which comprises 70% of the data, and a test set, which consists of the remaining 30%. This split allows for training and testing machine learning models.

4.5. Model Building

Several models have been constructed, and the dataset has been used to train and evaluate these models. The performance of each model has been thoroughly compared, and the best-performing model has been selected based on various evaluation metrics and criteria.

4.6. Save The Model

The selected model has been saved by converting it into a pickle file. This allows for easy storage and retrieval of the model for future use.

4.7. Cloud Setup & Pushing the App to The Cloud

AWS (Amazon Web Services) has been chosen as the deployment platform for the application. The application files have been loaded from the GitHub repository to the AWS environment, ensuring that the application is hosted and accessible on AWS infrastructure.

4.8. Application start & input data by user

The application has been initiated and is now ready for use. You can enter the required inputs into the application to perform the desired tasks.

4.9. Prediction

Once you've submitted the inputs, the application will execute the model and generate predictions. The output will be displayed as a message, providing information about whether the customer, whose demographic and behavioural data were entered as inputs, is likely to default in the following month or not.