**Low Level Design (LLD)**

**Credit Card Defaulters Prediction System**

|  |  |
| --- | --- |
| Written By | Subhajit Roy |
| Document Version | 1 |
| Last Revision Date | 06-Mar-2022 |

Document Control

Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1 | 06-Mar-2022 | Subhajit Roy |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviews:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Reviewer** | **Comments** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approval Status:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Review Date** | **Reviewed By** | **Approved By** | **Comments** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1. Introduction 1](#_Toc97549819)

[1.1. What is Low-Level design document? 1](#_Toc97549820)

[1.2. Scope 1](#_Toc97549821)

[1.3. Constraints 1](#_Toc97549822)

[1.4. Risks 1](#_Toc97549823)

[1.5. Out of scope 1](#_Toc97549824)

[2. Technical Specification 2](#_Toc97549825)

[2.1. Dataset 2](#_Toc97549826)

[2.1.1. Dataset Overview 2](#_Toc97549827)

[2.1.2. Input Schema 2](#_Toc97549828)

[2.2. Predicting Defaulters 3](#_Toc97549829)

[2.3. Logging 3](#_Toc97549830)

[2.4. Database 3](#_Toc97549831)

[2.5. Deployment 3](#_Toc97549832)

[3. Technology Stack 3](#_Toc97549833)

[4. Proposed Solution 4](#_Toc97549834)

[5. Model Training / Validation Workflow 4](#_Toc97549835)

[6. User I/O Workflow 4](#_Toc97549836)

[7. Exceptional Scenarios 5](#_Toc97549837)

[8. Test Cases 5](#_Toc97549838)

[9. Key Performance Indicators (KPI) 6](#_Toc97549839)

# Introduction

## What is Low-Level design document?

The purpose of this document is to present a detailed description of the Credit card default prediction System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to predict potential defaulters of the credit card before allotting it to a customer thereby, reducing the losses of the banks.

## Scope

This software system will be a Web application. The system will be designed to detect the credit card defaulters before allotting it to customer for better fund management. The system detects the defaulters based on information such as educational qualifications, age, past credit settlement amounts of last few months amount, previous repayment statues, etc.

## Constraints

We are considering an old dataset hence the latest trend of the defaulters is not captured in the model.

## Risks

Document specific risks that have been identified or that should be considered.

## Out of scope

Recommendation for the safe credit limit for the defaulters is out of scope for this project.

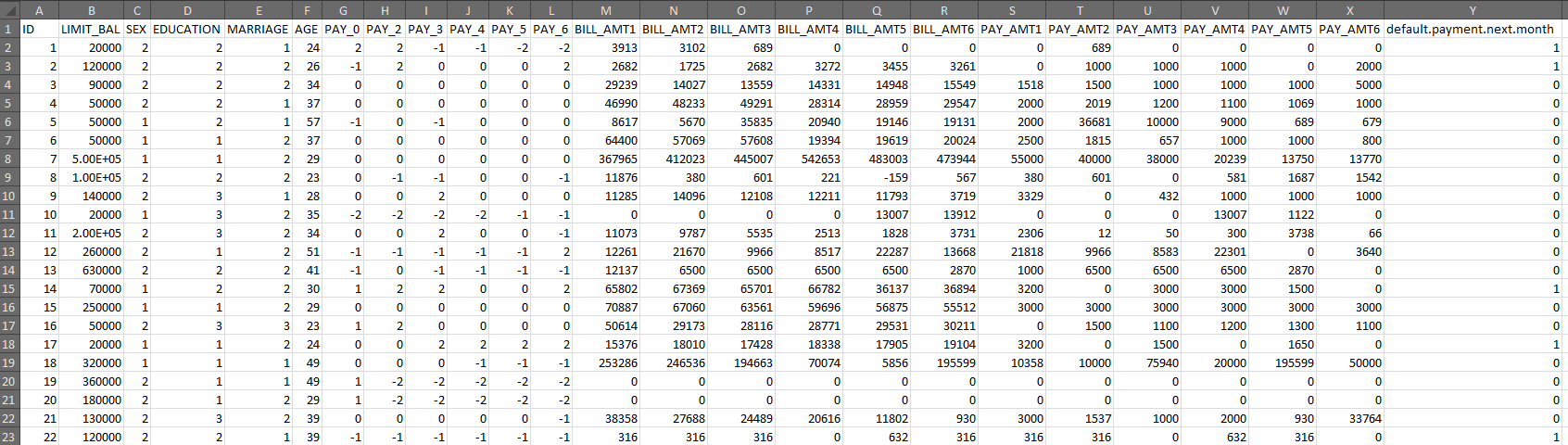
# Technical Specification

## Dataset

|  |  |
| --- | --- |
| Name | Default of Credit Card Clients Dataset |
| Description | Default Payments of Credit Card Clients in Taiwan from 2005 |
| Source | <https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset> |

### Dataset Overview

This dataset is a CSV file that 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.



### Input Schema

|  |  |
| --- | --- |
| **Feature Name** | **Feature Description** |
| 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 |
| ***Repayment Status:*** *(-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\_0 | Repayment status in September, 2005 |
| PAY\_2 | Repayment status in August, 2005 |
| PAY\_3 | Repayment status in July, 2005 |
| PAY\_4 | Repayment status in June, 2005 |
| PAY\_5 | Repayment status in May, 2005 |
| PAY\_6 | Repayment status in April, 2005 |
| 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) |

## Predicting Defaulters

The user will upload the file in a predefined CSV format containing the values of all featured variable. The file will be read and the model will provide the output in the form of CSV file which will be stored in S3 Bucket.

## Logging

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

* The System identifies at what step logging required.
* The System should be able to log each and every system flow.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

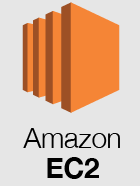
## Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

* The User uploads the training data using the web application.
* The user chooses the mode – further retrain the model or train the model from scratch
* Based upon the mode, either the existing data from the table will be deleted or the new data will be added to the existing table.

## Deployment

The model has been deployed on AWS EC2 instance running on Ubuntu platform. The model also uses S3 bucket to save and store the log and the output files.



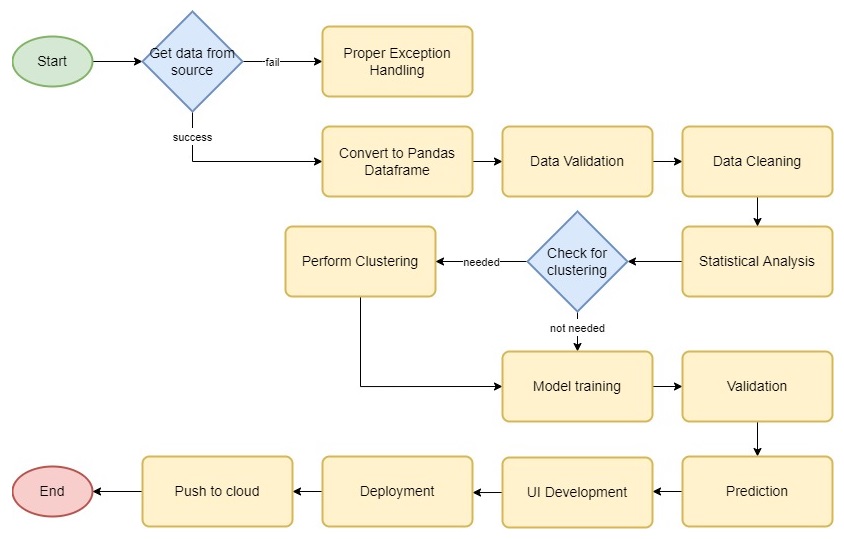
# Technology Stack

|  |  |
| --- | --- |
| Front End | HTML/CSS/JQuery |
| Backend | Python/Flask |
| Database | Cassandra |
| Deployment | AWS EC2 (Ubuntu Platform) |
| Storage | AWS S3 Bucket |

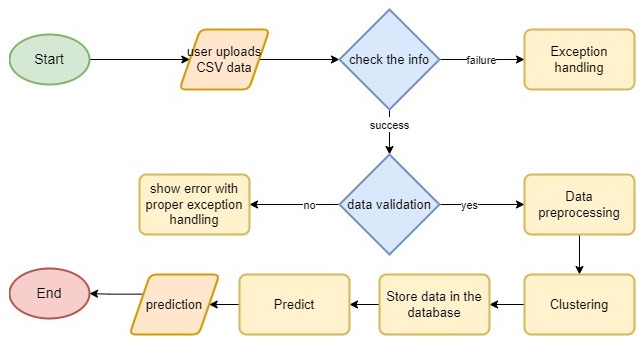
# Proposed Solution

The solution proposed here to build the application is to create separate models for different sets of data as a result of which the result will be aggregated and will provide a more accurate value. To separate the data into different sets, the data will be clustered and in each cluster various machine learning algorithms will be applied. The best performing model will be selected for that particular cluster. Similarly, during the prediction, the data will be clustered and in each cluster the already saved model will be applied to get the prediction.

# Model Training / Validation Workflow



# User I/O Workflow



# Exceptional Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Exception | Mitigation | Module |
| 06-Mar-2022 | 1.1 | First Draft | Subhajit Roy |
|  |  |  |  |

# 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 the user is able to upload the file for training | 1. Application is accessible  2. The browse button of the application is working properly | The user should be able to upload the file |
| Verify whether the model has been trained or not | 1. Application is accessible  2. The browse button of the application is working properly | The user should be able to see a success message at the end |
| Verify whether the user is able to upload the file for prediction | 1. Application is accessible  2. The browse button of the application is working properly | The user should be able to upload the file |
| Verify whether the prediction has been made or not | 1. Application is accessible  2. The browse button of the application is working properly | The user should be able to see a success message at the end |

# Key Performance Indicators (KPI)

* Credit Limit Balance
* Education
* Marriage
* Age
* Repayment status
* Bill settlement amount