

|  |  |  |
| --- | --- | --- |
| 05/07/2023 | Credit Card |  |
|  |  |
|  | Default Project |  |
|  | Architecture |  |

Yash Kumar Singh

Vaibhav Srivastava

CREDIT CARD DEFAULT PROJECT

Contents

|  |  |  |  |
| --- | --- | --- | --- |
| [Abstract](#page3) | | [.......................................................................................................................................](#page3) | [2](#page3) |
| [1](#page4) | [Introduction..........................................................................................................................](#page4) | | [3](#page4) |
|  | [1.1](#page4) | [Why this Low-Level Design Document? ...........................................................................](#page4) | [3](#page4) |
|  | [1.2](#page4) | [Scope ............................................................................................................................](#page4) | [3](#page4) |
|  | [1.3](#page4) | [Definitions .....................................................................................................................](#page4) | [3](#page4) |
| [2](#page5) | [Technical Specification ..........................................................................................................](#page5) | | [4](#page5) |
|  | [2.1](#page5) | [Dataset Overview...........................................................................................................](#page5) | [4](#page5) |
|  | [2.2](#page5) | [Predicting Credit Card Default.........................................................................................](#page5) | [4](#page5) |
|  | [2.3](#page5) | [Logging..........................................................................................................................](#page5) | [4](#page5) |
| [3](#page6) | [Technology Stack ..................................................................................................................](#page6) | | [5](#page6) |
| [4](#page6) | [Proposed Solution .................................................................................................................](#page6) | | [5](#page6) |
| [5](#page6) | [Workflow..............................................................................................................................](#page6) | | [5](#page6) |
| [6](#page7) | [Model Feature Importance ....................................................................................................](#page7) | | [6](#page7) |



1 | P a g e

CREDIT CARD DEFAULT PROJECT

Abstract

Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, one of the biggest threats faced by commercial banks is the risk prediction of credit clients. The goal is to predict the probability of credit default based on credit card owner's characteristics and payment history.

Use of various classification models like Decision Tree, Random Forest, XGBoost, MLPClassifier was done. XGBoost and CatBoost were selected as best models from above. The XGBoost model provided 82.63% accuracy in training. CatBoost model provided 81% of accuracy. Testing accuracy for model was 82.3%.

70% data was used for training and 30% data was used for testing.

After Providing various details the model will predict whether the transaction is fraud or not.



2 | P a g e

CREDIT CARD DEFAULT PROJECT

1 Introduction

1.1 Why this Low-Level Design Document?

The goal of LLD or a Low-level design document is to give an internal logical design of the actual program code for the Credit Card Default Probability Prediction. LLD describes the class diagrams with the methods and relations between classes and the 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 defined during data design work.

1.3 Definitions

|  |  |
| --- | --- |
| **Term** | **Description** |
| IDE | Integrated Development Environment |
| EDA | Exploratory Data Analysis |
| XGBoost | Extreme Gradient Boost Algorithm |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron Classifier |
| Classifier |  |
| KNN | K-Nearest Neighbours |
| VS Code | Visual Studio Code |



3 | P a g e

CREDIT CARD DEFAULT PROJECT

2 Technical Specification

2.1 Dataset Overview

For training and testing the model I used a publicly available dataset on Kaggle. The dataset contains features such as User, Card, Year, Month, Day, Amount, Use Chip, Merchant Name, Merchant City, Merchant State, Zip, MCC, Errors and output feature ‘is Fraud?’

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

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Variable Name** | **Measurement** | **Description** |
|  |  | **Unit** |  |
|  | LIMIT\_BAL | NT Dollar | Amount of given credit |
|  | SEX | Integer | Gender (1=male, 2=female) |
|  | EDUCATION | Integer | 1=graduate school, 2=university, 3=high |
|  |  |  | school, 4=others, 5=unknown, 6=unknown |
|  | MARRIAGE | Integer | Marital status (1=married, 2=single, |
|  |  |  | 3=others) |
|  | AGE | Years | Age of the person in years |
|  | PAY\_0-6 | Integer | Repayment status for various months |
|  | BILL\_AMT1-6 | NT Dollar | Amount of billed statements for various |
|  |  |  | months |
|  | PAY\_AMT1-6 | NT Dollar | Amount of Previous payments done |
|  | default.payment.next.month | Binary | Will Customer default? Yes/No |

2.2 Predicting Credit Card Default

The web application must be loaded properly for the users without any technical glitches like server timeouts.

* It must display the input fields and the “Predict” button to the users who accessed the application and allow the user to enter the values with respect to the attributes of the customer.
* The user gives the required information.
* Then the application should be able to predict Default and the probability of default based on the information given by the user about the customer.

2.3 Logging

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

* The system should be able to log every step in the program flow.
* System should not be hung even after using so many loggings.
* Logging makes debugging much easier, like we can directly go to that specific line of code, having bugs.
* In this project, logs will be written in the files “logfile.txt” in the log folder of filesystem.



4 | P a g e

CREDIT CARD DEFAULT PROJECT

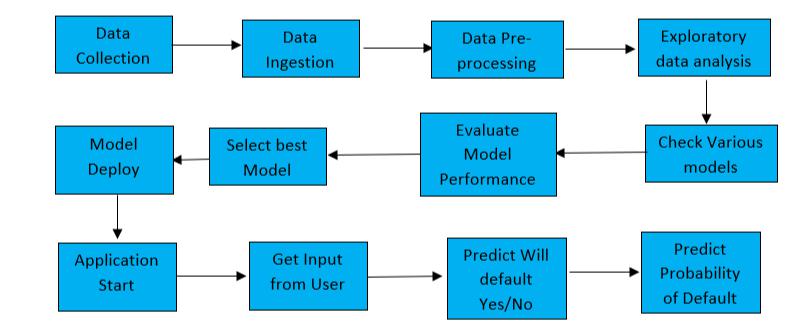
3 Technology Stack

|  |  |  |
| --- | --- | --- |
| Front End | Dash by Plotly |  |
| Back End | Python Version 3.10.7, |  |
| Dash Library Call-back decorator |  |
|  |  |
| Deployment | Heroku, Gunicorn |  |

4 Proposed Solution

The solution proposed here is a web application, which takes the details of the customer and those details will be taken by a machine learning model in the backend, which will then predict whether customer will default and the probability of default and display it on the front-end page of the user.

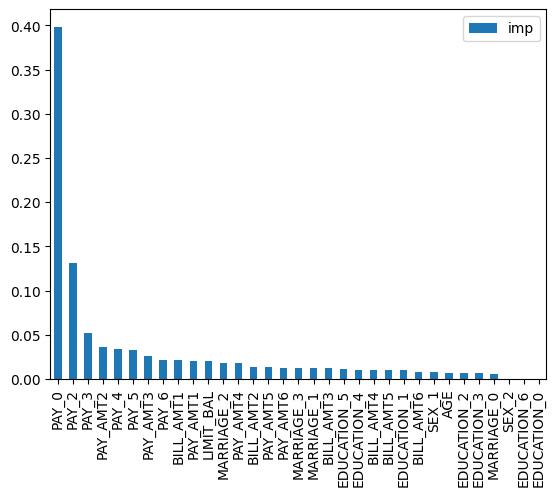
5 Workflow



5 | P a g e

CREDIT CARD DEFAULT PROJECT

6 Model Feature Importance



6 | P a g e