High Level Design (HLD)

##### **Credit Card Defaulter Prediction**

Revision Number: 1.0

Last date of revision: 05/02/2024

Shridatta Patil

**Contents**

[Document Version Control 3](#_Toc111624495)

[Abstract 4](#_Toc111624496)

[1 Introduction 5](#_Toc111624497)

[1.1 Why this High-Level Design Document? 5](#_Toc111624498)

[1.2 Scope 5](#_Toc111624499)

[2 General Description 6](#_Toc111624500)

[2.1 Problem Statement & Product Perspective 6](#_Toc111624501)

[2.2 Proposed Solution 6](#_Toc111624502)

[2.3 Data Requirements 6](#_Toc111624503)

[2.4 Tools used 7](#_Toc111624504)

[2.5 Constraints 8](#_Toc111624505)

[2.6 Assumptions 8](#_Toc111624506)

[3 Design Details 9](#_Toc111624507)

[3.1 Process flow 9](#_Toc111624508)

[3.2 Event log 9](#_Toc111624509)

[3.3 Error Handling 10](#_Toc111624510)

[3.4 Optimization 10](#_Toc111624511)

[3.5 Reusability 10](#_Toc111624512)

[3.6 Application compatibility 10](#_Toc111624513)

[3.7 Deployment 10](#_Toc111624514)

[4 Conclusion 11](#_Toc111624515)

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 05-01-2024 | 1.0 | First Version of Complete HLD | Shridatta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Abstract

This project addresses the issue of credit card default by developing a predictive model to identify high-risk cardholders. With the increasing prevalence of credit card use and associated defaults, financial institutions face significant challenges in managing credit risk. This project utilizes machine learning techniques to analyse historical data on credit card transactions, payment histories, and customer demographics. By leveraging algorithms such as logistic regression, decision trees, and many other algorithms, the model aims to accurately predict the likelihood of default, enabling proactive risk management strategies. The outcome of this project not only assists in minimizing financial losses but also enhances customer relationship management by identifying those who may benefit from targeted financial education and assistance. The project ultimately strives to improve the efficiency of credit risk assessment processes and contribute to the stability of financial institutions.

# Introduction

## Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) document is to add the necessary detail to the project description to represent a suitable model and coding for application. This document is also intended to help detect contradictions before coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of the design aspects and define them in detail
* Describe the user interface is implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:

o Security

o Reliability

o Maintainability

o Portability

o Reusability

o Application compatibility o Resource utilization

o Serviceability

## Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology stack. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

# General Description

## Problem Statement & Product Perspective

The dataset contains Card Limit Balance, Education, Age, Sex, Marital Status, Bill amounts and Payments made by the card holder.

* + - To detect the card defaulter.
    - To create API interface to predict the outcome.

## Proposed Solution

The solution proposed here is to determine the defaulter based on the given parameters. At first analyze how the gender, marital status and sex affect the prediction. In the second case, the card limit balance, payments made and bill amount will affect the prediction. And in the last use case, we will be making an interface to predict the result.

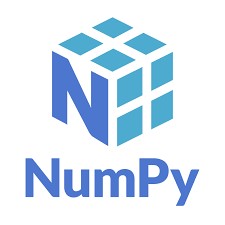
## Data Requirements

Data requirements completely depend on out problem statement.

* Comma separated values (CSV) file.
* Input file feature/field names and its sequence should be followed as per decided.

## Tools used

* Python programming language.
* Libraries such as Pandas, NumPy, Scikit-Learn, Matplotlib
* Framework: Flask
* IDE: Jupyter Notebook, VSCode.
* Cloud service: Heroku
* GitHub





* VSCode and Jupyter notebook is used as IDE.
* For Visualization of the plots Matplotlib is used.
* Front-end development is done using HTML/CSS.
* Python Flask is used for backend development.
* GitHub is used for version control system

## Constraints

MLOPs on the cloud must be fully automated in consideration of continuous integration, continuous deployment with retraining approach of model, and archiving the data over time. Users can easily use the application and not needed to know any of the workings.

## Assumptions

The main objective of the project is to develop an API to predict the card defaulter on the basis of the parameters given. Machine learning based classification model is used for predicting above mentioned cases on the input data.

# Design Details

## Process flow

Data Modelling

Model Testing

Data Ingestion

Start

Flask Setup

Feature-engineering

Data Validation

Data Transformation

Deployment

Figure 1: Process flow

## Event log

The system should log every event so that the track of every detail will be known and what process is running currently could be seen.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging is required.
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## Error Handling

The system should identify the errors encountered; an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

## Optimization

Data strategy derives performance:

1. Filling missing values.
2. Replacing outliers.
3. Creating new features from cut expenses feature.
4. Hyperparameter tuning
5. Validating score again

## Reusability

The code written and the components used should have the ability to be reused with no problems.

## Application compatibility

The different components for this project will be using Python as an interface between them. Each component will have its task to perform, and it is the job of Python to ensure the proper transfer of information.

# Conclusion

In this project, the system shows us that the different techniques that are used in order to determine the credit card defaulter on the basis of given parameters. Also, significant difference between payments and bill amounts. Accuracy, which plays a key role in prediction-based system. From the results we could see that Gradient Boosting turned out to be best working model for this problem in terms of the accuracy. Our predictions help user to know which card holder will become defaulter from their usage of the card.