HIGH LEVEL DESIGN (HLD)

CREDIT CARD DEFAULT PREDICTION

Document Version Control

Date Issued	Version	Description	Author
23/08/2023	1	HLD-V1.0	Nagalakshmi M

Contents

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 faces 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.

1. Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. 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 being 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: -

- Security
- Reliability
- Maintainability
- Portability
- Reusability
- Application compatibility
- Resource utilization
- Serviceability

1.2 Scope

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

2. General Description

2.1 Product Perspective

The goal of this project is to predict the probability of the Credit card Defaulter, based on the information of default payments, demographic factors, credit data, history of payment and bill statement and many others of the credit card clients .

2.2 Problem Statement

To achieve the goal, we used a data set that is formed by taking into consideration some of the information of 30000 credit card holder in Taiwan in which the dataset contains the transaction details 30000 card holders from April 2005 to September 2005.The problem is based on the given information about each individual we have to calculate that whether that individual with Defaulter or not a defaulter for next month

2.4 Tools used

Business Intelligence tools and libraries works such as NumPy, Pandas, Seaborn, Matplotlib, Scipy, Sklearn, Jupyter Notebook and Python Programming Language are used to build the whole framework.

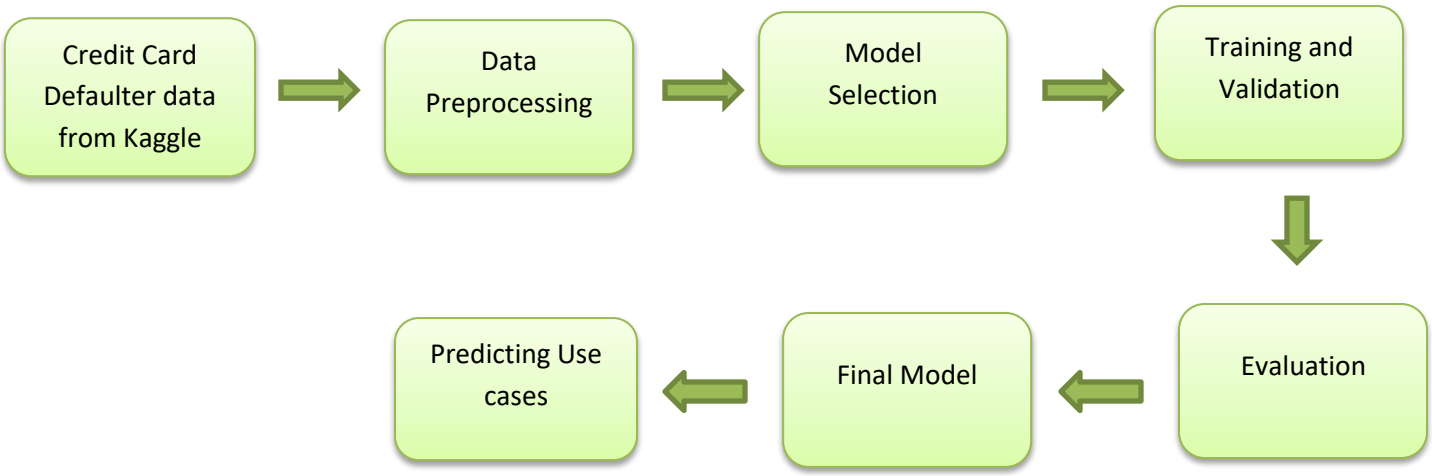
- Spyder IDE
- Visualization in matplotlib,seaborn
- Streamlit for backend development

3. Design Details

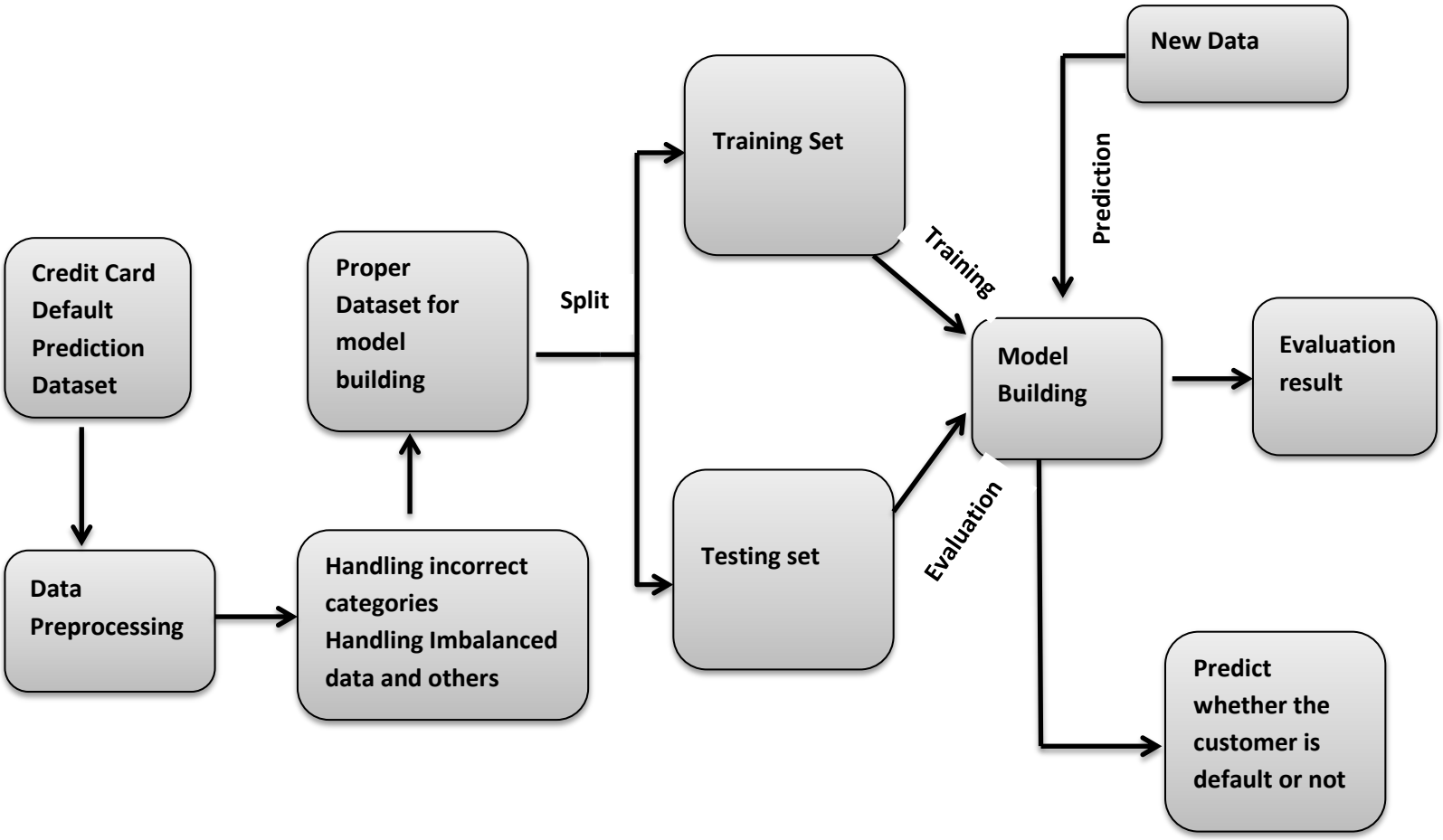
3.1 Process Flow

To predict the credit card defaulter of the next month we used Machine Learning Algorithm to build the model. To identify the different types of tags, we will use the machine learning model for classification. Below is the process flow diagram is as shown below.

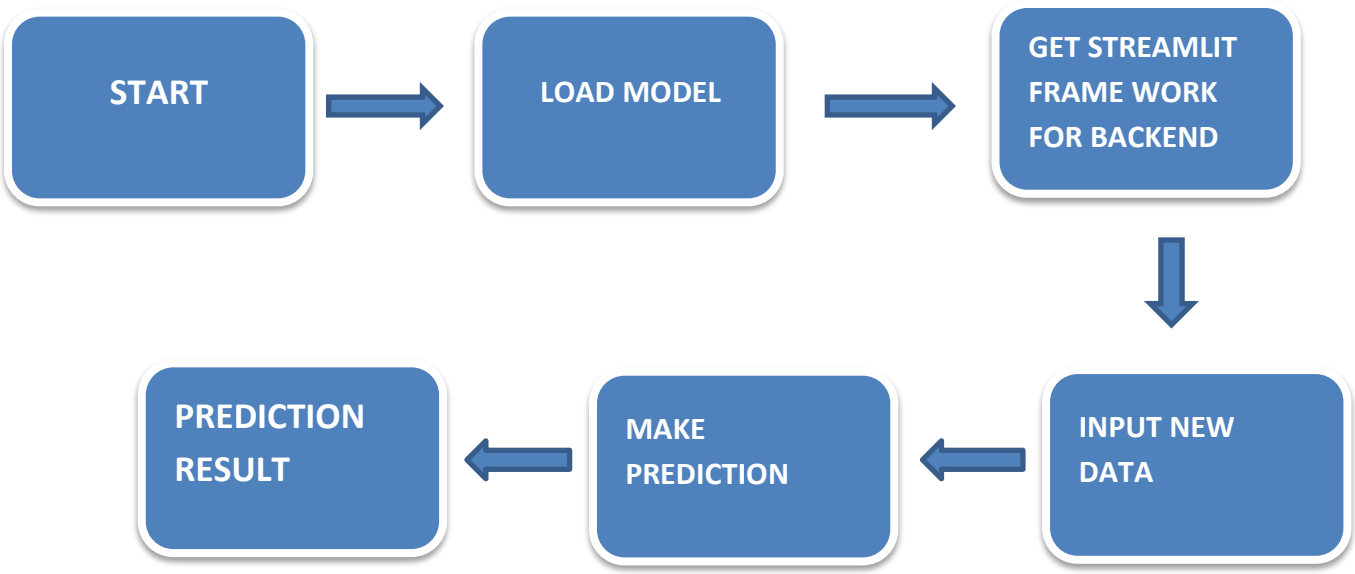
Proposed Methodology



3.1.1 Model Validation And Evaluation



3.1.2. Deployment Process



3.2 Event Log

The system should log every event, so the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.

2. The System should be able to log each system flow.

3. Developer can choose the 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 too.

3.3 Error Handling

Should errors be 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.

4. Performance

The Classification Algorithm is used is used to identify whether the customer is defaulter or not by providing the details of 6 months transaction. In the final model we concluded with Random Forest which combination many decision tree where the result is quite good in predicting the result.

4.1 Reusability

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

4.2 Application Compatibility

The different components for this project will use 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.

5. Conclusion

The Classification Algorithm is used is used to identify whether the customer is defaulter or not by providing the details of 6 months transaction.