HIGH LEVEL DESIGN (HLD)

CREDIT CARD DEFAULT PREDICTION

G.SANDEEP

25/12/2023

Table of Contents

ABSTRACT

Credit risk plays a major role in the banking industry business. Banks' main activities involve granting loans, credit cards, investments, mortgages, and others. The credit card has been one of the most booming financial services by banks over the past years. However, with the growing number of credit card users, banks have been facing an escalating credit card default rate. As such data analytics can provide solutions to tackle the current phenomenon of managing credit risks. This project discusses the implementation of a model which predicts if a given credit card holder has a probability of defaulting in the following month, using their demographic data and behavioral data from the past 6 months. Credit Card Default Prediction

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 prior to 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 o Reliability o Maintainability

- Portability o 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

From a product perspective, developing a User Credit Card Default Prediction system involves considering various elements to create a valuable and effective tool for financial institutions. Here's a breakdown of the product perspective for a Credit Card Default Prediction system:

- User Value Proposition
- User Interface and User Experience (UI/UX)
- Customization and Adaptability
- Model Interpretability
- Real-Time Monitoring and Alerts
- Feedback Mechanism

2.2 Problem statement

In response to the increasing demand for robust credit risk management in the financial sector, there is a need for a web-based application that predicts the probability of credit card default for customers. The solution aims to leverage demographic data and behavioral patterns from the past six months to provide accurate and timely insights into credit risk. The absence of such a system poses challenges to financial institutions in effectively assessing and mitigating credit risk, potentially leading to financial losses. Therefore, the goal is to develop a user-friendly web application that empowers financial analysts and risk managers with a predictive tool for making informed credit decisions.

2.3 Proposed Solution

The proposed solution is a web application for predicting the probability of credit card default based on demographic data and behavioral data from the previous six months, you can further refine the problem statement to include details specific to the development and implementation of this web application

2.4 Further Improvements

This project can also feature certain improvements like:
With the above feature, we could also add a feature that allows the user to access a dashboard, which will provide you with an overall analysis of the uploaded data (for example, the percentage and distribution of customers who paid their dues in the previous months, distribution of age of customers, etc.).We can also formulate a score on how much confidence the model has output produced. This feature can be implemented on the current project, as well as with the improvements mentioned above. A feature which allows the application to segment those customers which the model has predicted to default, based on a given criterion/criteria, which will allow the business team to make targeted strategies on each segmented groups.

2.5 Technical Requirements

To implement the web-based Credit Card Default Prediction System, you'll need to define specific technical requirements. Here's a list of key technical requirements that would contribute to the successful development and deployment of the system:

- Data Integration
- Predictive Model
- Web Application Framework
- Real-Time Updates
- User Authentication and Authorization
- Scalability

2.6 Data Requirements

This dataset is taken from kaggle(url:https://www.kaggle.com/uciml/defaultof-credit-card-clients-dataset).It 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
  - -1: Paid in full;
  - 0: No consumption;
  - 1 = payment delay for one month;
  - 2 = payment delay for two months; . . .;
  - o 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

2.7 Tools Used

Python programming language and frameworks such as NumPy, and Pandas, Scikit-learn is used to build the whole model.

- Jupyter Notebook is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- Streamlit CloudApp is used for deployment of the front-end development is done using Stremlit
- Python is used for backend development.
- GitHub is used as version control system.

2.7.1 Hardware Requirements

- Windows 11 Operating System
- Quad-core processor (e.g., Intel Xeon or AMD Ryzen)
- 16 GB RAM or more
- SSD Storage

2.7.2 ROS (Robotic Operating System)

"Robot Operating System (ROS) is an open-source robotics middleware suite. Although ROS is not an operating system, it functions as a collection of software frameworks for robot software development. It provides services specifically designed for a heterogeneous computer cluster, offering features such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management."

2.8 Constraints

when developing a Credit Card Default Prediction system, several constraints must be considered. Constraints can be limitations or conditions that may impact the design, development, and deployment of the system. Here are common constraints associated with Credit Card Default Prediction systems:

- Data Privacy and Compliance
- Interpretability and Explainability
- Imbalanced Data:

2.9 Assumptions

In the context of developing a Credit Card Default Prediction system, various assumptions are often made to simplify the modeling process and guide decision-making. However, it's essential to recognize that these assumptions may not always hold true in every real-world scenario. Here are some common assumptions associated with Credit Card Default Prediction:

- Stationarity of Data
- Independence of Observations
- Normality of Residuals
- Linear Relationship:

3. Design Details

3.1 Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram is as shown below.

3.1.1 Model Training and Evaluation

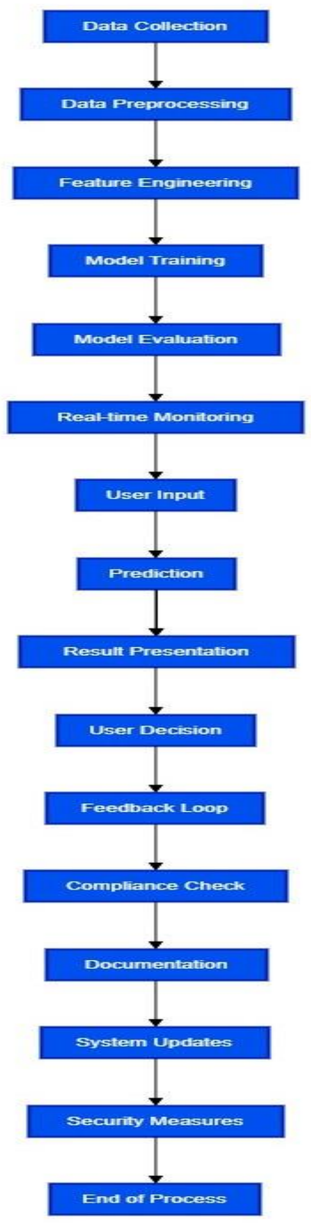

Figure 3.1.1: Model Training and Evaluation Diagram

3.1.2 Deployment Process

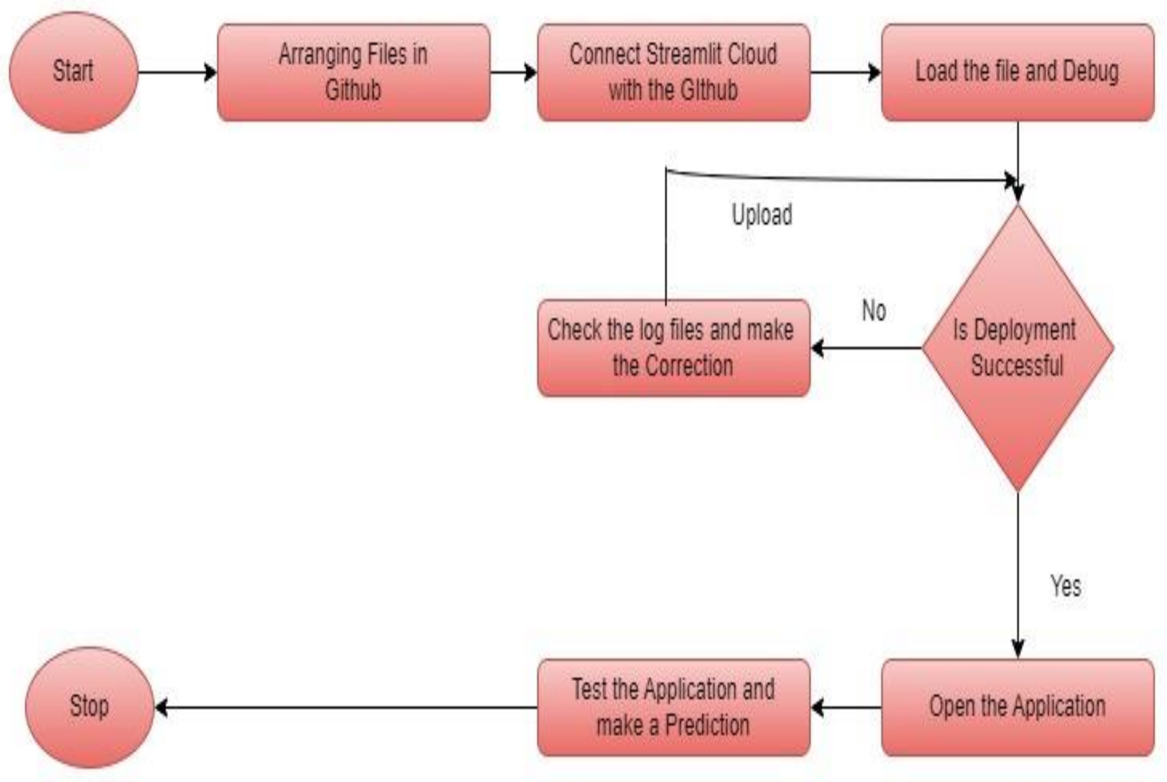
Start
Arranging Files in Github
Connect Streamlit Cloud with the GIthub
Load the file and Debug
Upload
Check the log files and make the Correction
No
Is Deployment Successful
Yes
Stop
Test the Application and make a Prediction
Open the Application

Figure 3.1.2: Deployment Process

3.2 Event log

An event log is a chronological record of events that occur within a system, application, or process. Each entry in the log typically includes information such as the time of the event, a description of the event, and relevant details about the context in which it occurred. Event logs are commonly used for monitoring, troubleshooting, auditing, and analyzing the behavior of systems.

3.3 Error Handling

Error handling is a critical aspect of software development that involves managing and responding to unexpected situations or errors that may occur during the execution of a program. Proper error handling helps improve the reliability, robustness, and user experience of software applications.

3.4 Performance

The Credit Card Default Prediction App is used to predict whether a given customer is likely to default in the following month or not based on the customer's demographic data and behavioral data for the previous N number of months(in this, project, we consider N=6). This will allow the business institution to take note and strategize the next step accordingly (wrt the given customer).

3.5 Reusability

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

3.6 Application Compatibility

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

3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.8 Deployment

The model can be deployed in any cloud services such as Microsoft Azure, AWS, Google, Streamlit Cloud etc

4. Conclusion

This application will predict whether a given customer will likely default or not in the following month, based on various demographic and behavioral data, and can help financial institutions sin taking necessary actions before the event occurs.