CREDIT CARD DEFAULT PREDICTION

: High Level Design (HLD) :

CREATED BY

Rakesh

Document Version Control

Date	Version	Description	Author
25-July-2022	1.0	Initial HLD	Rakesh

Table of 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 faced by commercial banks is the risk prediction of credit clients. The goal is to predict the probability of credit defaults 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 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
  - ✓ Reliability
  - ✓ Maintainability
  - ✓ Portability
  - ✓ Reusability
  - ✓ Application
  - ✓ compatibility

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.

1.3 Definitions

Term	Description
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
Heroku	Platform where deployed the application.
CDP	Credit default prediction

2. General Description:

2.1 Product Perspective

The Credit card default Prediction system is a machine learning-based default prediction model which will help us to know the possibility to default credit card payment by a customer.

2.2 Problem statement

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.

2.3 PROPOSED SOLUTION

The solution proposed here is a Credit card default Prediction model can be implemented to perform above mention use case. In this case, we have to enter the last six months bill amounts , paid amounts, payment status, credit limit, personal details. Based on the above details model predicts 0 or 1. 0 means the person won't default ,1 means default.

2.4 Technical Requirements

This document addresses the requirements for detecting Credit default possibility of a customer based on his transaction history. We require a good working system with very good internet connection to use the app. As we are deploying the application in Heroku we can access the app using its link. So, the above will be sufficient.

2.5 Data Requirements

Data requirement completely depend on our problem statement.

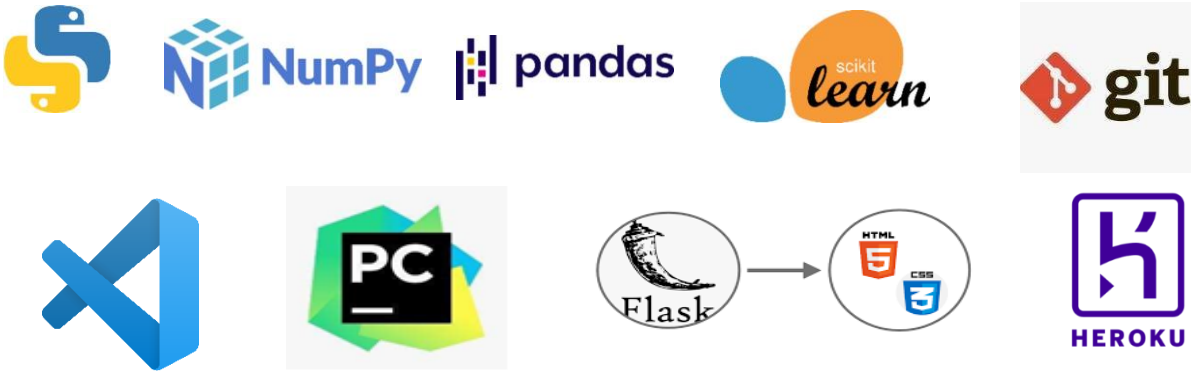
- We need at least 30000 datapoints.
- Each datapoint must include customer details, payment statuses, bill amounts, payment history
- Amount of given credit in NT dollars includes individual and family/supplementary credit.
- Gender of the customer.

- Educational status weather he/she is graduate/school etc.…
- Marital status of the customer.
- Age of the customer.
- Payment status of last six months.(For how long he/she paid the credit back.)
- Bill amounts of the customer for the last six months.
- Paid amount against the bills for last six months.
- Whether the customer defaulted the payment of next month or not(0=not defaulted, 1=Defaulted).

2.6 Tools used

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

- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- Heroku is used for deployment of the model.
- Cassandra is used to retrieve, insert, delete, and update the database.
- Front end development is done using HTML/CSS.
- Python is used for backend development.
- GitHub is used as version control system.
- Vs Code is used as IDE



2.7 Constraints

The CDP application must be user friendly, as automated as possible and users should not be required to know any of the workings.

2.8 Assumptions

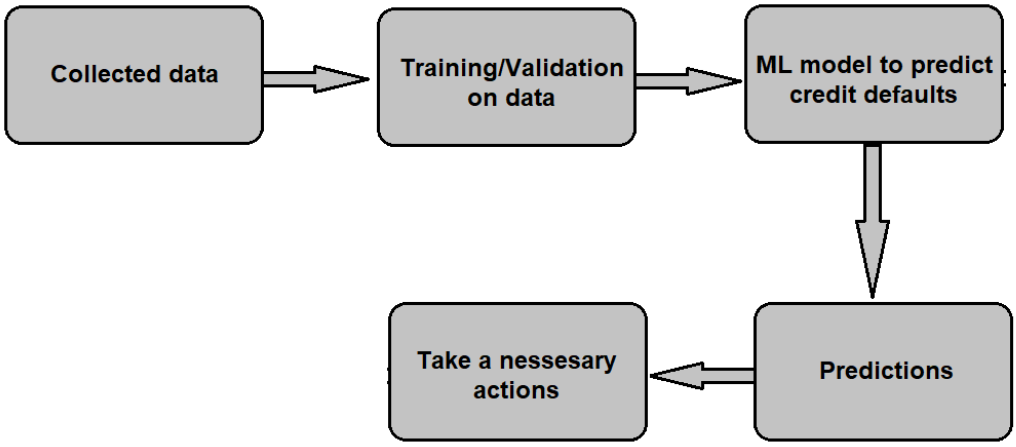
The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes through forms in application webpage. Machine Learning based prediction model is used for detecting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in the designer is expecting.

3. Design Details

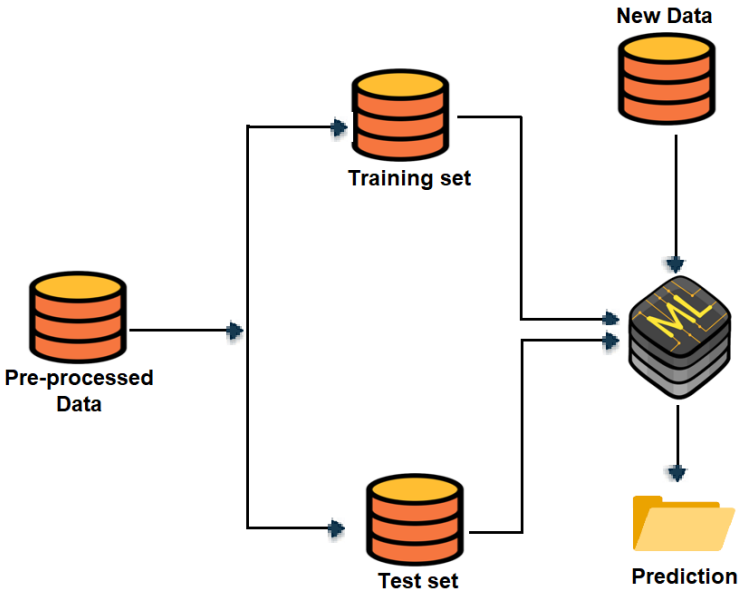
3.1 Process Flow

For predicting the possibility to default, we will use a machine learning base model. Below is the process flow diagram is as shown below.

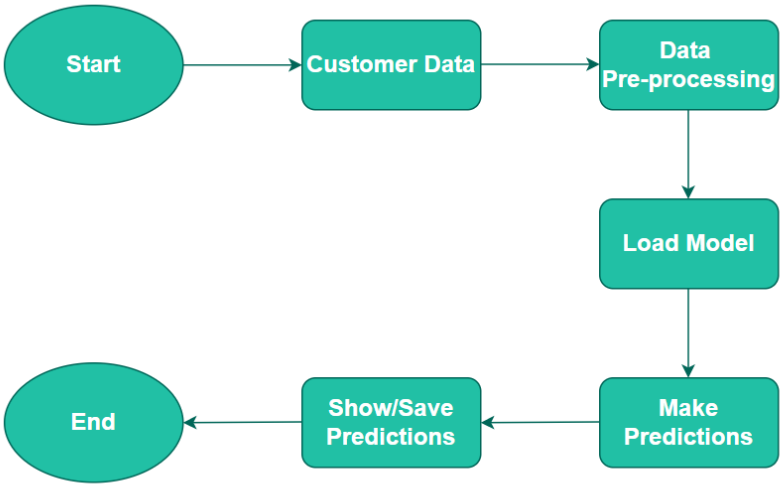
Proposed methodology



3.2 Model Training and Evaluation



3.3 Deployment Process



3.4 Event Logs:

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

Initial Step-By-Step Description:

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

3.5 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 CDP application is used for detection of probability of credit default by a credit card user. As we already trained the model with sufficient data it predicts the default with high accuracy. Also, model retraining is very important to improve the performance.

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 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.

4.3 Resource Utilization

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

4.4 Deployment

We will be deploying the application in Heroku .To deploy the app to Heroku, we use the git push command to push the code from local repository's master or main branch to our Heroku remote.

5. Conclusion

The CDP model will predict possibility of a customer to default next month payment based on various customer credit card usage data used to train our algorithm, so we can identify the possible behavior of a customer in early stages and can take necessary action to avoid defaulting the payment by the customer, so we can have a strong business in banking.

THE END