

Credit Card Default Prediction

High Level Design

Report by:
Snehal Kharade
Abhishek Nimje
07 May, 2023

Document Version Control

Date Issued	Description	Author
07 May 2023	Initial HLD	Abhishek Nimje
12 May 2023	Reviewed HLD	Snehal Kharade

Contents

Document Version Control	2
Abstract.....	4
1 Introduction	5
1.1 Why this High-Level Design Document?	5
1.2 Scope	5
1.3 Definitions.....	5
2 General Description.	6
2.1 Product Perspective	6
2.2 Problem statement	6
2.3 Proposed Solution.....	6
2.4 Further Improvements	6
2.5 Data Requirements	7
2.6 Tools used	8
3 Design Details.....	10
3.1 Process Flow.....	10
3.2 Model Training and Evaluation	10
3.2.1 Deployment Process	11
3.3 Event log	11
3.4 Error Handling	11
3.5 Performance	12
3.6 Reusability.	12
3.7 Application Compatibility.....	12
3.8 Resource Utilization.....	12
3.9 Deployment	12
4 Conclusion.....	14

Abstract

There are times when even a seemingly manageable debt, such as credit cards, goes out of control. Loss of job, medical crisis or business failure are some of the reasons that can impact your finances. In fact, credit card debts are usually the first to get out of hand in such situations due to hefty finance charges (compounded on daily balances) and other penalties. A lot of us would be able to relate to this scenario. We may have missed credit card payments once or twice because of forgotten due dates or cash flow issues. But what happens when this continues for months? How to predict if a customer will be defaulter in next months? To reduce the risk of Banks, this model has been developed to predict customer defaulter based on demographic data like gender, age, marital status and behavioral data like last payments, past transactions etc.

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.

1.2. 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:
 - o Security
 - o Reliability
 - o Maintainability
 - o Portability
 - o Reusability
 - o Application compatibility
 - o Resource utilization
 - o Serviceability

1.3. 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 Credit Card Default Predictor is a machine learning-based classification model which will help us to predict credit card defaulters and take the necessary action.

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 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.

2.3. Proposed Solution

The solution proposed here is a web application, which predicts the probability of credit default for a customer based on the customer's demographic data and behavioral data from previous months (6 months).

2.4. Further Improvements

This project can also feature certain improvements like:

- Including a feature, allowing the user to upload a data file on customers and predict the possibility of credit card default for each of the customers. The file uploaded should have columns in the same format as that of the data used for training the model.
- With the above feature, we could also add a feature which allows the user to access a dashboard, which will provide you an overall analysis of the uploaded data (example, 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 on the 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. Data Requirements

This dataset is taken from kaggle([url: https://www.kaggle.com/uciml/defaultof-credit-card-clients-dataset](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:

Feature Name	Feature Information
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, 4=others, 5=unknown, 6=unknown)
MARRIAGE	Marital status (1=married, 2=single, 3=others)
AGE	Age in years
PAY_0	Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 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)

2.6. Tools used

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

- Jupyter Notebook is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- HerokuApp is used for deployment of the model.
- Front end development is done using HTML/CSS
- Python is used for backend development.
- GitHub is used as version control system.

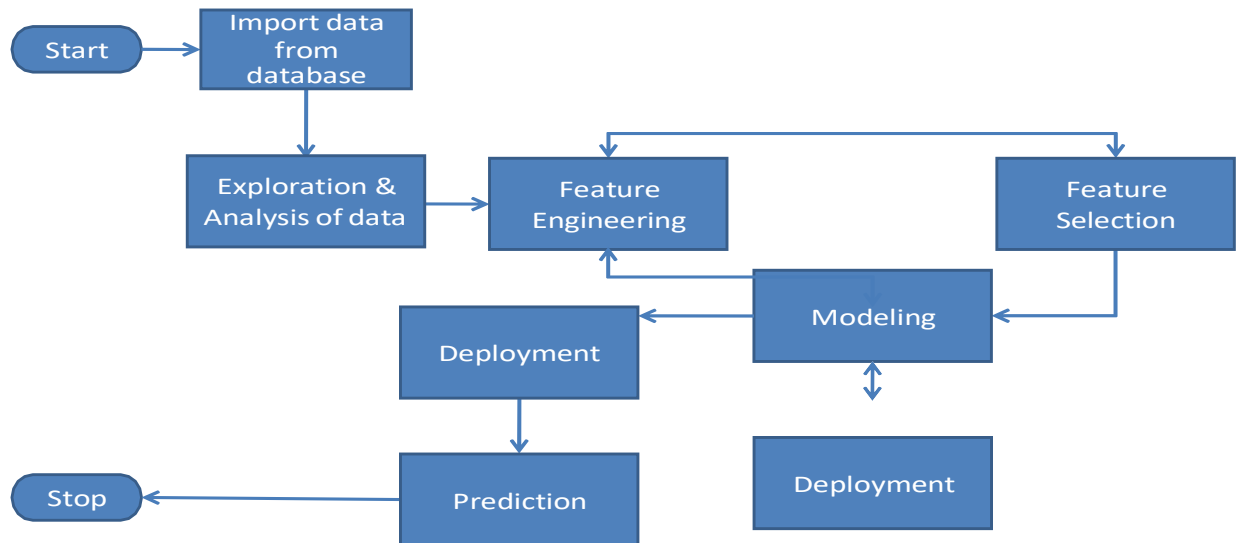


3. Design Details

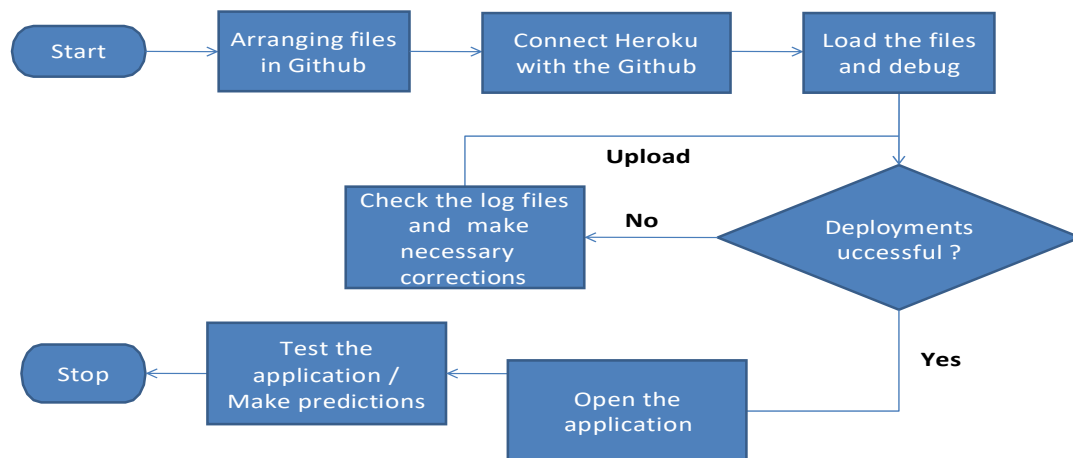
3.1. Process Flow

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

3.2. Proposed methodology



3.3. Deployment Process



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 (w.r.t. given customer).

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

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

5. 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 in taking necessary actions before the event occurs.