CREDIT CARD DEFAULT PREDICTION

Revision Number: 1.0
Last date of revision: 28/04/2022

Document Version Control

Date Issued	Version	Description	Author
28/04/2022	1	Initial HLD — V1.0	Poonam Kanojia

Contents

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.

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.

1.3 Definitions

Term	*Description*
CCD	Credit Card Default
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
Heroku	Cloud Platform

2 General Description

2.1 Product Perspective

This model predicts whether the customer will be defaulter in the next month or not by implementing various machine learning tasks.

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 risen. In this way, one of the biggest threats faced by commercial banks is the risk prediction of credit client's. The goal is to predict the probability of credit default based on credit card owner's characteristics and payment history.

2.3 PROPOSED SOLUTION

Model predicts whether the customer will be defaulter in the next month or not by implementing various machine learning tasks and applying different classification algorithm and then selecting best model by tuning hyper parameters and then deploying the model using flask on Heroku platform.

2.4 Data Requirements

1. ID: ID of each client
2. LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary = credit)
3. SEX: Gender (1=male, 2=female)
4. EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
5. MARRIAGE: Marital status (1=married, 2=single, 3=others)
6. AGE: Age in years
7. 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)
8. PAY_2: Repayment status in August, 2005 (scale same as above)
9. PAY_3: Repayment status in July, 2005 (scale same as above)
10. PAY_4: Repayment status in June, 2005 (scale same as above)
11. PAY_5: Repayment status in May, 2005 (scale same as above)
12. PAY_6: Repayment status in April, 2005 (scale same as above)
13. BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
14. BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
15. BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
16. BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)

17. BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
18. BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
19. PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
20. PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
21. PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
22. PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
23. PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
24. PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
25. default.payment.next.month: Default payment (1=yes, 0=no).

2.5 Tools used

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

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

2.6 Constraints

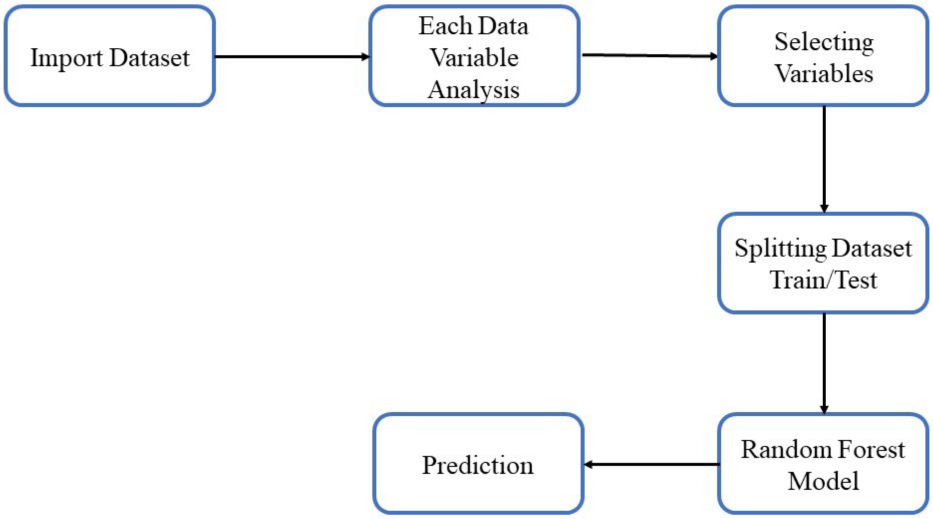
The credit card default prediction model must be user friendly, as automated as possible and users should not be required to know any of the workings.

3 Design Details

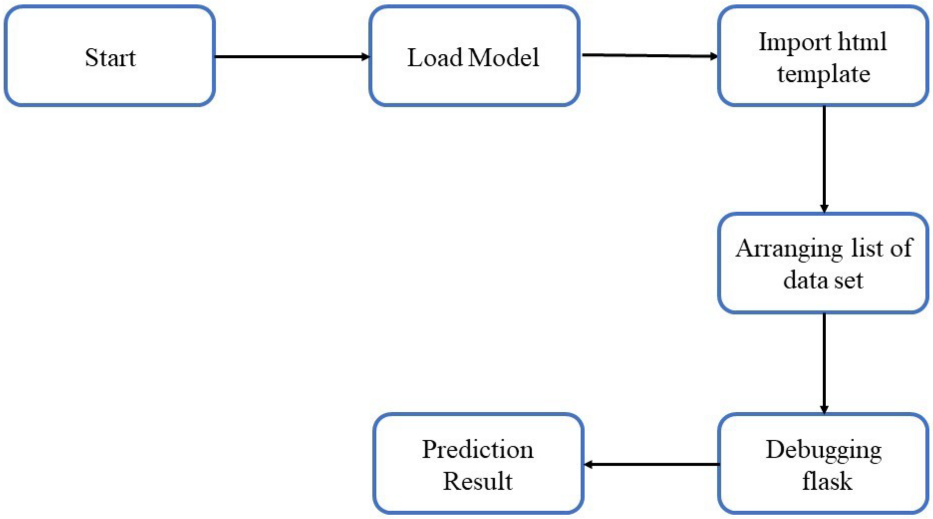
3.1 Process Flow

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.

Proposed methodology



3.1.1 Deployment Process



3.2 Event log

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

Initial Step-By-Step Description:

1. The System identifies at what step logging 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.

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

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

Deployment is done using Heroku cloud platform.

5.0 CONCLUSION

The project is designed in flask; hence it is accessible to everyone. The above designing process will help banks and loan lenders predict whether customers will default the credit card payment or not, so the bank or respective departments can take necessary action, based on the model's predictions. The UI is made to be user-friendly so that the user will not need much knowledge of any tools but will just need the information for results.

6.0 References

[1] J. Stavinset al., “Credit card borrowing, delinquency, and personal bankruptcy,”NewEngland Economic Review, pp. 15–30, 2000.

[2] J. A. Nelder and R. W. Wedderburn, “Generalized linear models,”Journal of the RoyalStatistical Society: Series A (General), vol. 135, no. 3, pp. 370–384, 1972.

[3] R. M. Neal,Bayesian learning for neural networks. Springer Science & Business Media,2012, vol. 118