High Level Design (HLD)

CCDP (Credit Card Default Prediction)

Revision Number: 1.0
Last date of revision: 05/07/2023

Document Version Control

Date Issued	Version	Description	Author
05/07/2023	1	Initial HLD - VI .0	Gurudev Singla

Contents

Abstract

To build a classification methodology to determine whether a person defaults the credit card payment for the next month. Banks have to face a lot of loss due to defaulters of credit card payment. In banking defaulters are those who takes money from bank but don’t return back. So the basic aim is to build an application which will be able to predict the probability of happening default. So that bank will avoid to make looses due to credit card payment.

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
CCDP	Credit Card Default Prediction
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment

2 General Description

2.1 Product Perspective

The CCDP solution system is a machine learning-based classification model which will help us to get the probability of occurring default for the next month payment.

2.2 Problem statement

To build a classification methodology to determine whether a person defaults the credit card payment for the next month.

2.3 PROPOSED SOLUTION

The solution proposed here is an CCDP based classification model that can be implemented to perform above mention use cases. If the probability of default is very high then bank will take some action on the customer or try to avoid giving credits to those customers.

2.4 FURTHER IMPROVEMENTS

CCDP can be added with more use cases like prediction for giving loans to the customers of bank, it will give the probability of default in returning loan to the bank and many other domains as well.

2.5 Technical Requirements

This document addresses the requirements for detecting default. But there is no such technical requirements.

2.6 Data Requirements

Data requirement completely depend on our problem statement.

- We need data to train our model.
- We require dataset which contains record of customer did they default or not.
- The format of file should be :

```
Attr
    "SampleFileName": "creditCardFraud_021119920_010222.csv",
    "LengthOfDateStampInFile": 8,
    "LengthOfTimeStampInFile": 6,
    "NumberofColumns" : 25,
    "ColName": {
        "ID" : "int",
        "LIMIT_BAL" : "float",
        "SEX": "int",
        "EDUCATION": "int",
        "MARRIAGE": "int",
        "AGE": "int",
        "PAY_0": "int",
        "PAY_2": "int",
        "PAY_3": "int",
        "PAY_4": "int",
        "PAY_5": "int",
        "PAY_6" : "int",
        "BILL_AMT1": "float",
        "BILL_AMT2": "float",
        "BILL_AMT3": "float",
        "BILL_AMT4": "float",
        "BILL_AMT5": "float",
        "BILL_AMT6": "float",
        "PAY_AMT1": "float",
        "PAY_AMT2": "float",
        "PAY_AMT3": "float",
        "PAY_AMT4" : "float",
        "PAY_AMT5": "float",
        "PAY_AMT6": "float",
        "default.payment.next.month": "int"
```

- The file should be in csv format.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask API, json file, Cassandra, postman are used to build the whole solution.

CASSANDRA, FLASK API, POSTMAN

PyCharm is used as IDE.

- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- No cloud platform is used to deploy because it needs credit card.
- Cassandra 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 for keeping project.

2.7.1 Hardware Requirements

- PC

2.8 Constraints

The CCDP classification solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

2.9 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 CCD which will be taken from bank. Machine Learning based classification 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 way a designer is expecting.

iNeuron

3 Design Details

3.1 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.

3.2 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 CCDP classification bases solution is used for prediction of default customers in the society whenever CCDP predicts probability of customer to be default it will return a file and that file will be seen by authorities of bank and takes necessary action, so it should be as accurate as possible.

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 are not deploying this solution on cloud because we don't have credit card.

4

5 USER INTERFACE

First you have to use postman app and try to hit the local host port 5001 by giving the file path that is training csv file path on successful training of model it will return you "Training Successful !!!" kind of message after that for prediction you have to open your web browser and try to hit the local host port 5001 then if you want custom file to be predicted then you have to pass the file path where file is located and if you want default file to be predicted then you just have to click on "Default file predict" option after predicting successfully it will return the path where prediction output file is located.

Conclusion

The solution proposed here is an CCDP classification based model that can be implemented to predict the default customers. If the probability of default is very high then bank will take some action on the customer or try to avoid giving credits to those customers. So that bank will able to avoid the loss.

References

1. https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset