Credit Card Default Prediction

Revision Number: 2.0

Last date of revision: 20/01/2024

Document Version Control

Date Issued	Version	Version	Author
8/12/2023	1	Initial HLD — V1.0	Aditya Patil
20/1/2024	2	Final HLD __V2.0	Aditya Patil

Contents

Abstract

This project focuses on creating a predictive model for commercial banks to assess credit default risk among their clients. The approach involves standard machine learning tasks, including Data Exploration, Data Cleaning, Feature Engineering, Model Building, and Model Testing. By experimenting with various machine learning algorithms tailored to credit card owner characteristics and payment history, the goal is to provide a reliable solution for predicting the probability of credit default.

The anticipated outcome is a practical tool that enables commercial banks to make informed decisions about credit approvals, leveraging insights from client data. This project's simplicity lies in its adherence to fundamental machine learning principles, making it accessible and applicable for banks seeking a straightforward yet effective solution to manage credit risk.

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 the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the 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.

2 General Description

2.1 Product Perspective

The Credit Card Default Prediction is a machine learning based model for helping banks predict whether the bank customer will default on the credit card bill next month or not.

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

To predict credit default risk, we'll start by exploring the data, understanding features related to credit card owners and payment history. We'll clean the data by handling missing values and outliers. Next, we'll create useful features and standardize data for consistency. Using machine learning algorithms like logistic regression and decision trees, we'll build models and fine-tune them for accuracy. Testing the models on a separate dataset will ensure they predict credit default probabilities effectively. Through an iterative process, we'll refine the models and deliver a simple yet reliable tool for commercial banks to make informed decisions about credit approvals.

2.4 Data Requirements

- **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)

2.5 Tool Used

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

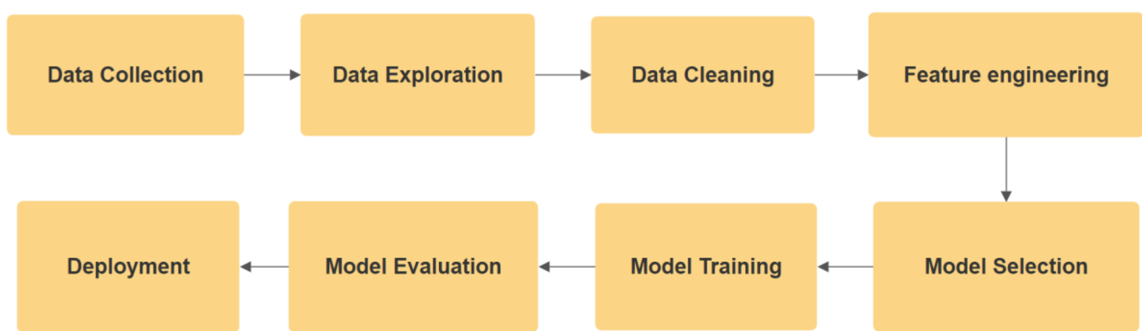
- Visual Studio Code is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn is used.
- Tableau/Power BI is used for dashboard creation.
- Front end development is done using HTML/CSS
- Python Flask is used for backend development.
- GitHub is used as a version control system.

3 Design Details

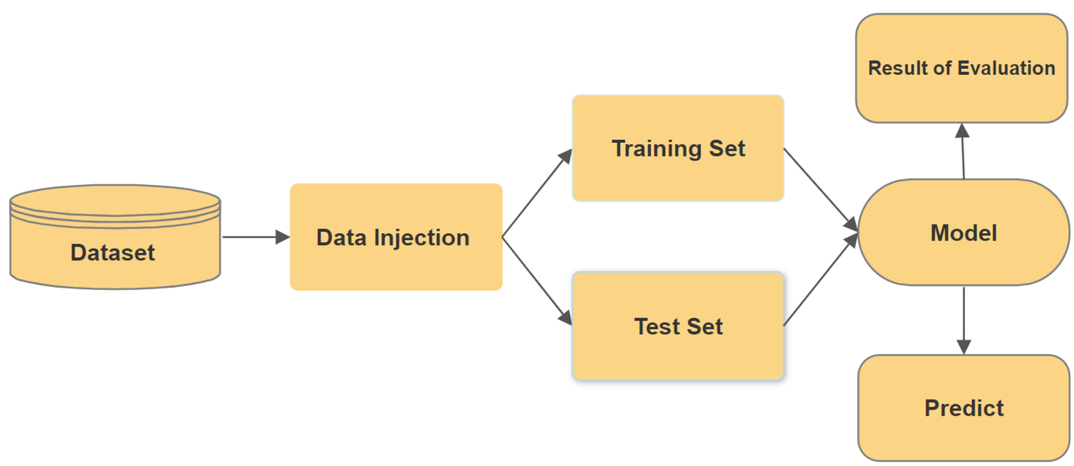
This detailed design outlines the step-by-step process, ensuring a systematic and thorough approach to building and deploying the credit default prediction model.

3.1 Process Flow

Proposed methodology



3.1.1 Model Training and Evaluation



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 every system flow.
3. Developers can choose logging methods. 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

The plan we have should work well for predicting if someone might have trouble paying back a credit. We're going step by step, making sure our data is good, creating useful features, and using smart computer techniques. We're picking the best methods that are simple to understand and will help banks make better choices about giving credit. We'll test it a lot to make sure it works in different situations, keeping things straightforward and practical for the banks to use. Our goal is to make a tool that's accurate and easy to apply in the real world.

4.1 Reusability

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

4.2 Accuracy

The accuracy of the project's predictive model for credit default risk is determined by how well it correctly identifies cases of credit default. Through the systematic application of machine learning techniques, data exploration, and model training, we aim to achieve a high level of accuracy in predicting whether a credit card owner is likely to default.

4.3 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.

6 Conclusion

In conclusion, this project endeavors to provide commercial banks with a practical and accurate tool for predicting credit default risk. By systematically navigating through data exploration, cleaning, and feature engineering, we have crafted a model that leverages machine learning algorithms tailored to credit-related characteristics. The focus on key performance metrics, such as accuracy, precision, and recall, ensures that the model is not only effective but also reliable in real-world scenarios. The iterative refinement process and comprehensive testing contribute to the model's robustness and generalizability. This project aims to empower commercial banks with a straightforward solution, enhancing their ability to make well-informed decisions about credit approvals and effectively manage credit risk.