High Level Design (HLD)

CREDIT CARD DEFAULTER PREDICTION

Document Version Control

DATE	DESCRIPTION	AUTHOR
20-06-2023	HLD- Version 1.0	SHREYASH VIRENDRA CHAWDA

• Introduction

Machine Learning Project for predicting credit card defaulter prediction. Based on history of last 6 months model it’s trying to predict customer next month payment status which helps financial organization for their repayment of the amounts and also helps to minimize the burden of financial assets.

• 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
- Scope

The HLD documentation presents the structure of the system 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.

• Product Perspective

The credit card defaulter is Machine Learning based model in which it’s predicting next month customer status of defaulter or not.

• 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.

• PROPOSED SOLUTION

The solution is Credit card defaulter prediction model in which its predicting customer condition based on their last six months payment information with the help of last six months behavior and approach toward clearing debt model will predict next month’s status as defaulter or will clear the debt.

• FURTHER IMPROVEMENTS

Credit card defaulter model in further improvements may need more data while interlinking its different banks and income status which will helps to predict. The credit score which will be given by organization and its reviews and feedbacks given can be used as sentiments analysis and that parameter may help us to predict more accuracy.

• Data Requirements

Data requirement completely depend on our problem statement.

There are 25 independent variables (including `id`):

* 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)

- Tools used

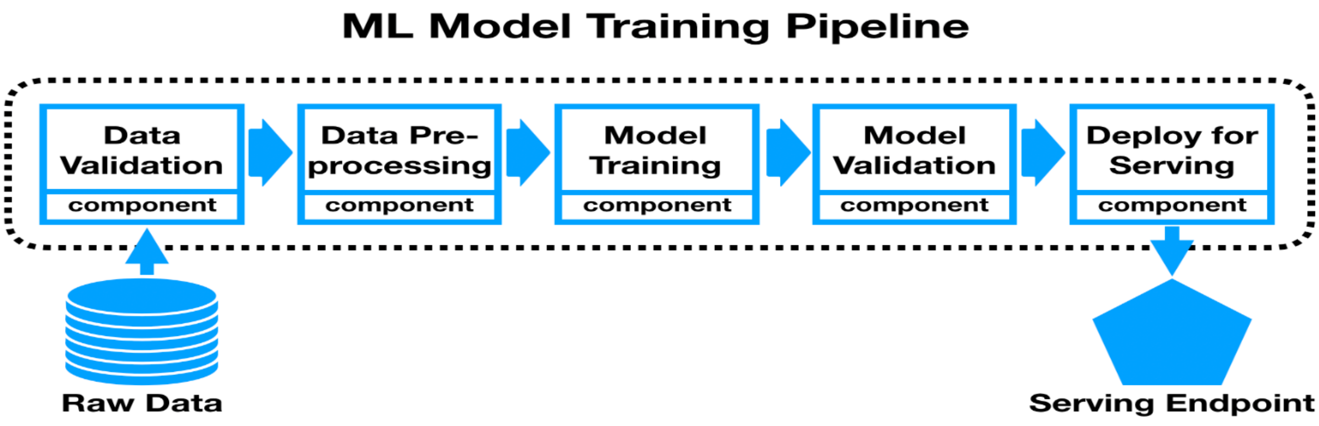
1. VS code is used as IDE.
2. Pandas, Numpy is used for data wrangling.
3. Scikit learn library is used for pipeline and model building.
4. For visualization of the plots, Matplotlib, Seaborn are used.
5. Docker is used for deployment.
6. Front end development is done using HTML.
7. GitHub is used as version control system.

- Design Details

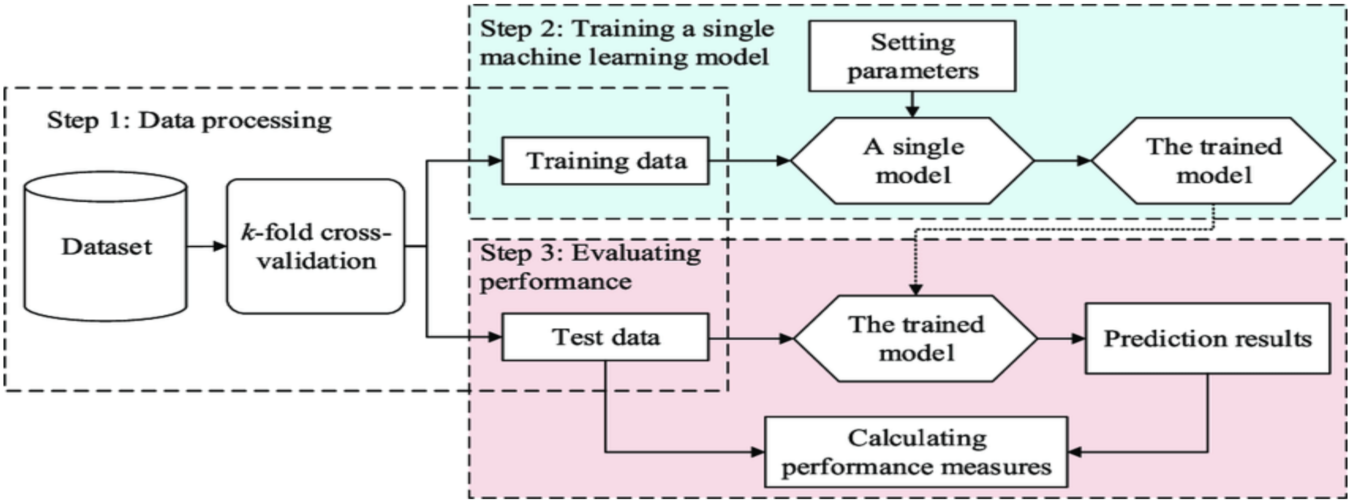
Process Flow

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

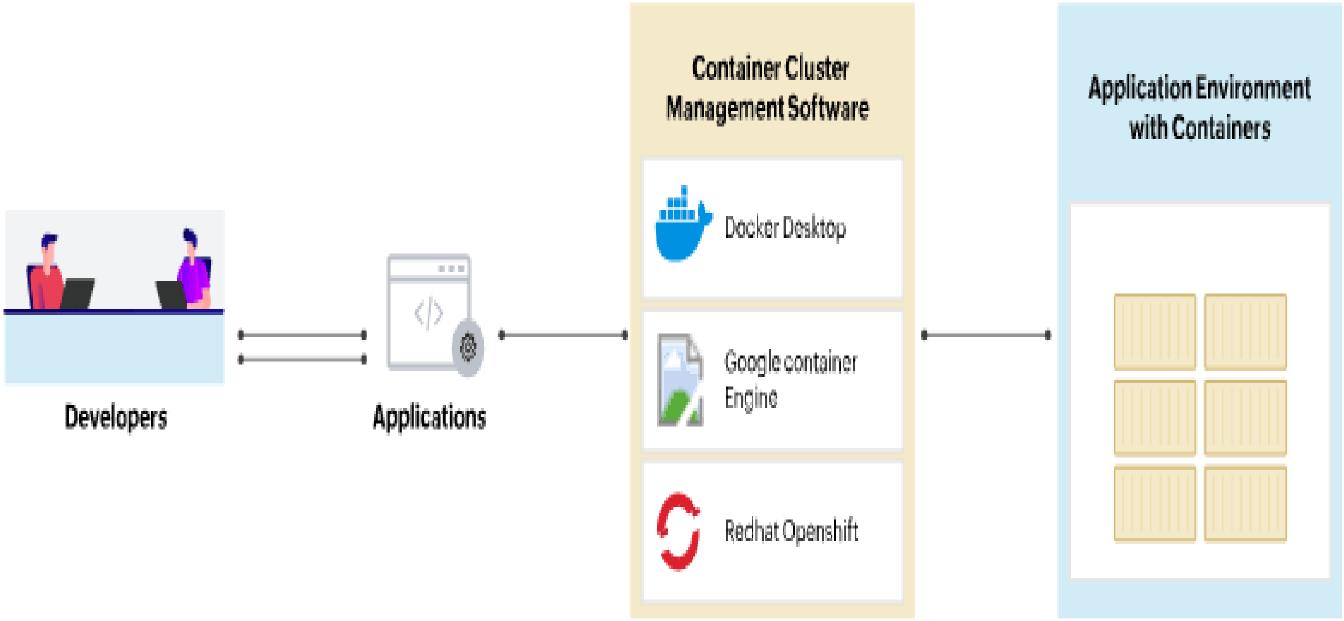
Proposed Model



- Model Training and Evaluation



- Model Deployment


Container Cluster Management Software
Application Environment with Containers
Docker Desktop
Google container Engine
Redhat Openshift
Developers
Applications

- Event log

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.

- 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.

- Performance

The credit card defaulter prediction model gives accuracy of 80 percent in predicting based on parameter given. Also different parameters like customers income details and there different bank account if exists then the feedback of that customer from banks required to give more reliable and robust result.

- Reusability

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

- 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.

- Resource Utilization

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

- KPIs (Key Performance Indicator)

1. Keep the track and predict that customer will be defaulter.
2. Helps financial organization from future losses.
3. Helps in growing businesses by understanding customers.
4. Customer who are not defaulter financial corporate may provide more service and facility who strengthen the relation
5. Minimize the financial burden to the corporate.