HIGH LEVEL DESIGN (HLD)

CREDIT CARD DEFAULT

CONTENT

DOCUMENT VERSION CONTROL …….…………………………………………………………………………………………………2

ABSTRACT …………………………………………………………………………………………………………………………………………4

1. INRODUCTION………………………………………………………………………………………………………………………5
   1. WHY THIS HIGH LEVEL DESIGN DOCUMENT ……………...…………………………………………………..5
   2. SCOPE ………………………………………………………………………….....................................................5
   3. DEFINATION …………………………………………………………………………………………………………………..5
2. GENERAL DISCRIPTION…………………………………………………………..................................................6
   1. PRODUCT PERSPECTIVE ………………………………………………………………………………………………….6
   2. PROBLEM STATEMENT …………………………………………………………………………………………………..6
   3. PROPOSED SOLUTION ……………………………………………………………………………………………………6
   4. FURTHER IMPROVEMENTS……………………………………………………………………………………………..6
   5. TECHNICAL REQUIREMENTS …………………………………………………………………………………………..6
   6. TOOLS USED……………………………………………………………………………………………………………………6
   7. DATA REQUIREMENTS…………………………………………………………………………………………………….7
   8. CONSTRAINTS ………………………………………………………………………………………………………………..7
   9. IMPROVEMENTS …………………………………………………………………………………………………………….7
3. DESIGN DETAILS ……………………………………………………………………………………………………………………7
   1. PROCESS FLOW ………………………………………………………………………………………………………………7
      1. PROPOSED METHODOLOGY ………………………………………………………………………………7
      2. MODEL TRAINING AND EVALUATION…………………………………………………………………8
      3. MODEL DEPLOYMENT APPROARCH …………………………………………………………………..9
   2. EVENT LOG ……………………………………………………………………………………………………………………9
   3. ERROR HANDLING ………………………………………………………………………………………………………….9
4. PERFORMANCE……………………………………………………………………………………………………………………..9
   1. REUSEABLITLITY………………………………………………………………………………………………………………9
   2. APPLICATION COMPATIBILTY …………………………………………………………………………………………9
   3. RESOURCE UTILIZATION ……………………………………………………………………………………………….10
   4. DEPLOYMENT ………………………………………………………………………………………………………………10
5. KEY PERFORMANCE INDICATOR …………………………………………………………………………………………10
6. CONCLUSION………………………………………………………………………………………………………………………10

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issue | Version | Description | Author |
| 21/01/2022 | 1 | Initial HLD – V 1.0 | Swapnil Sonawane |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

ABSTRACT

**Credit card default** is the term used to describe what happens when a credit card user makes purchases by charging them to their credit card and then they do not pay their bill. It can occur when one payment is more than 30 days past due, which may raise your interest rate. Most of the time, the term default is used informally when the credit card payment is more than 60 days past due. A default has a negative impact on the credit report and most likely lead to higher interest rates on future borrowing.

**1. INTRODUCTION**

* 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 DESGIN ASEPCTS AND DEFINE THEM IN DETAIL
* DESCRIBE THE USER INTERFACE BIENG IMPLEMENTED
* DESCRIBE THE HARDWARE AND SOFTWARE INTERFACE
* DESCRIBE THE PERFORMANCE REQUIREMENT
* INCLUDE DEFINE FEATURE AND ARCHITECHTURE OF THE PROJECT
* LIST AND DESCRIBE THE NON-FUNCTIONAL ATTRIBUTES
* SECURITY
* RELIABLILTY
* MAINTAINABILITY
* PORTIBILITY
* REUSEABILITY
* APPLICATION COMPATIBILTY
* 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. DEFINATION

|  |  |
| --- | --- |
| TERM | DESCRIPTION |
| DATABASE | Collection of Information Monitored by the System |
| IDE | Integrated Development Environment |
| AWS | Amazon Web Services |

**2. GENERAL DISCRIPTION**

2.1 PRODUCT PERSPECTIVE

Credit Card Default is UI based application which will be used for predicting the defaulter based on their previous record.

2.2 PROBLEM STATEMENT

To create an UI application which can be used to detect whether the the person is defaulter or not?

2.3 PROPOSED SOLUTION

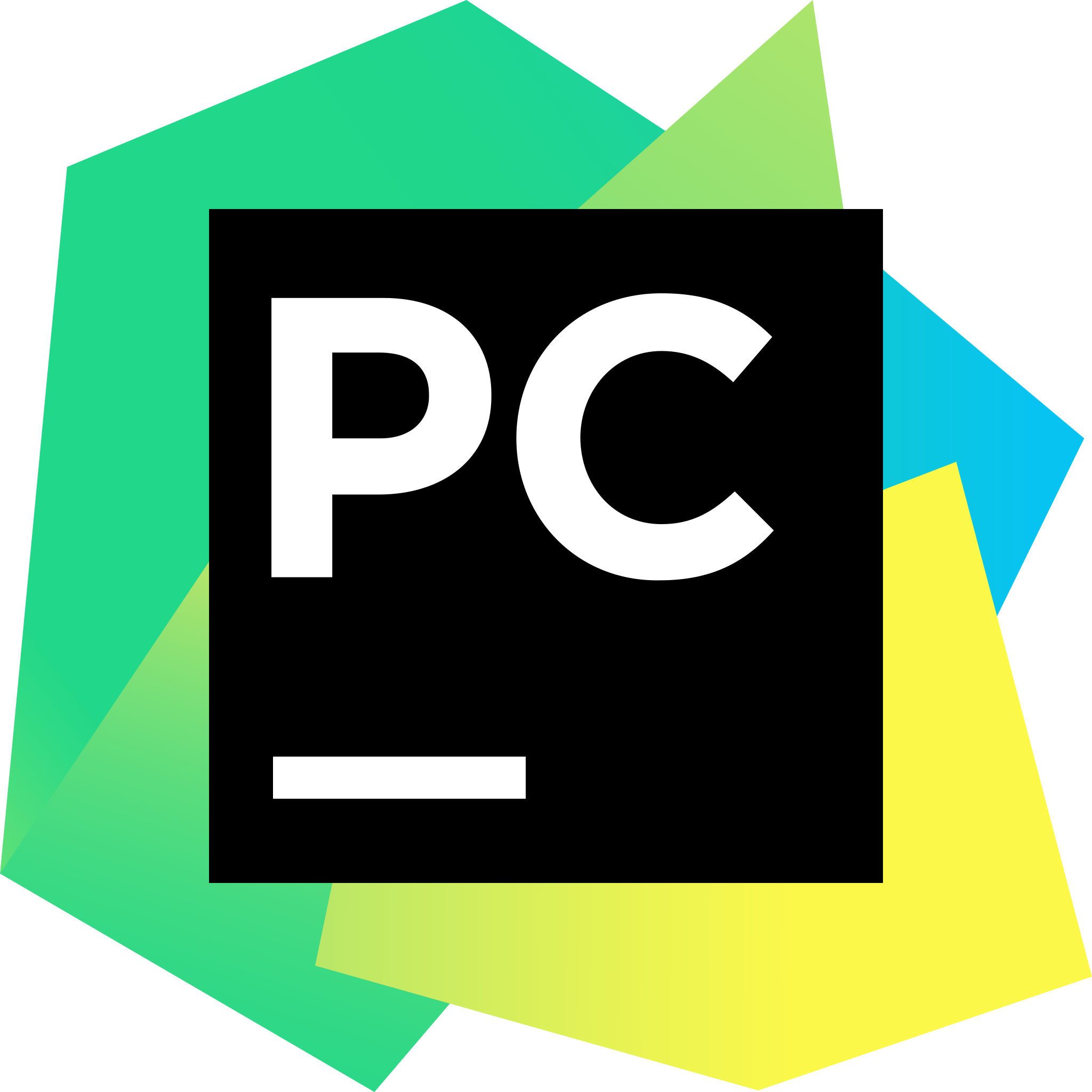
The solution to the above stated problem is creating a UI application. The UI will enable the user to check that the person is defaulter or not.

2.4 FURTHER IMPROVEMENTS

More advancement can be done for better results.

2.5 TECHINICAL REQUIREMENTS

No hardware tool is required but services like cloud services are required to host the website and database is required to store the data.

2.6 TOOLS USED



2.7 Data Requirements

* For training and testing the model, we are using Credit Card Fraud dataset from Kaggle
* From the user we are taking the following input.
  + 1. Limit\_Bal
    2. Sex
    3. Education
    4. Marriage
    5. Age
    6. Pay\_0 to Pay\_6
    7. Bill\_Amt1 to Bill\_Amt6
    8. Pay\_Amt1 to Pay\_Amt6

2.8 CONSTRAINTS

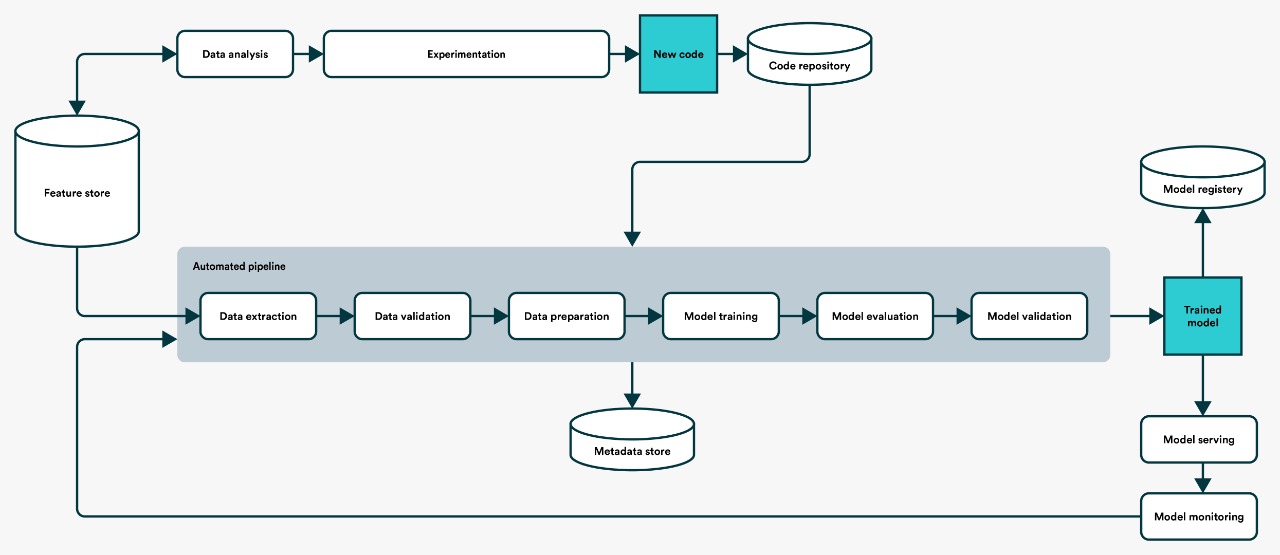
The Credit Card Fraud Detection System must be user friendly, errors free and users should not be required to know any of the back end working.

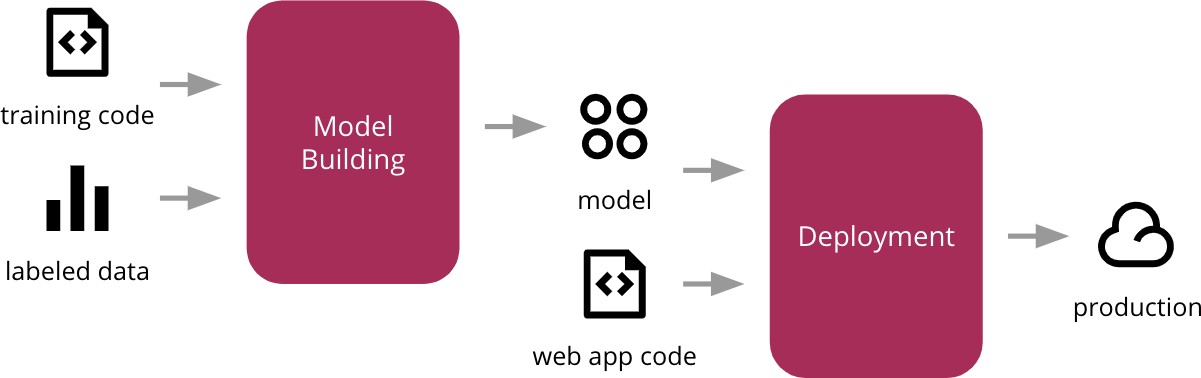
2.9 ASSUMPTIONS

The application will assume data to be in .csv format which comma ( , ) as a separator.

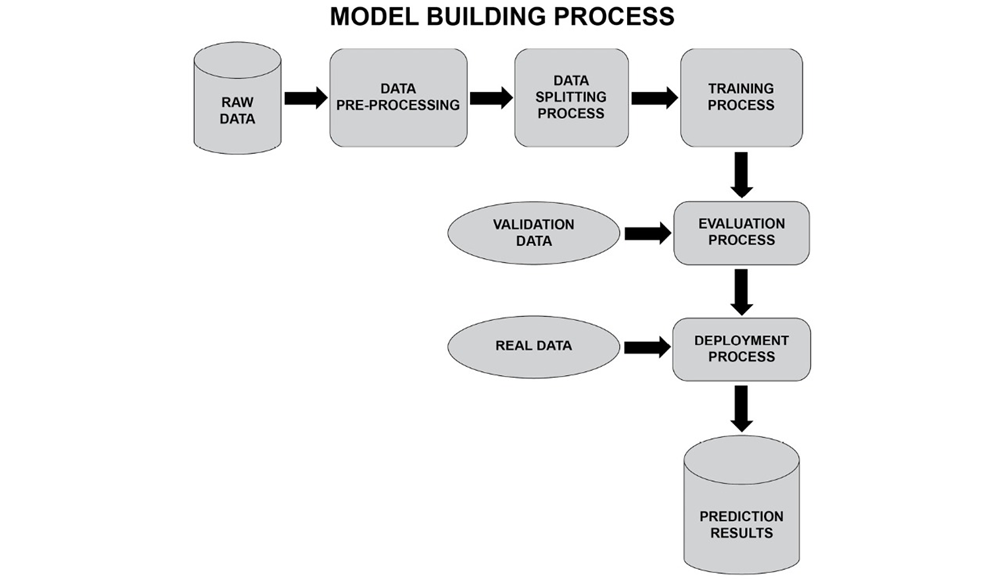
3. DESIGN DETAIL

3.1 PROCESS FLOW



3.1.2 Model training and evaluation

3.1.3 Deployment Process



3.2 EVENT LOG

The system is going to log everything so that the user gets to know which process is running internally.

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 hosted website will be used by many daily professionals so coding will be done in a proper modular fashion to reduce the run time and for faster execution.

4.1 REUSABLILITY

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

4.2 APPLICATION COMPATIBILTY

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 RESOUCE UTILIZATION

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

4.4 DEPLOYMENT

5. KEY PERFORMANCE INDICATOR

The only indicator in our application will be the accuracy of the application. This means that whether the application is able to predict correct output for a given input.

6. CONCLUSION

The application will be providing the banks an interactive platform where they just have to give some required inputs, and based to previous data, model will be able to produce a prediction for credit risk.