Low Level Design (LLD)

Low Level Design (LLD)

Credit Card Default Prediction

Revision Number – 1.0

Last Date of Revision – 16-05-2023

**ARYA  DIXIT**

1

Low Level Design (LLD)

2

Low Level Design (LLD)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | | | | | | | | | | | |  |  |  |  |
| [**Abstract**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.gjdgxs) | |  | | | | | | | | | | | | | | [4](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.gjdgxs) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**INTRODUCTION**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.30j0zll) | | | | |  | | | | | | | | | | | [5](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.30j0zll) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**Why this LLD documentation?**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1fob9te) | | | | | | | | | | | | | | | [5](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1fob9te) | | |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**1 Architecture**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3znysh7) | | | | | | | | | | | | | | | | [**3**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3znysh7) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| [**2 Architecture Design**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3dy6vkm) | | | | | | | |  | | | | | | | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3dy6vkm) | | | |
|  |  | | |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  | [**2.1 Data Gathering from Main Source**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1t3h5sf) | | | | | | | | | | | | | | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1t3h5sf) | | |  |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |
|  | [**2.2 Tools Used**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.17dp8vu) | | | |  | | | | | | | | | | |  | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.17dp8vu) | |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |
|  | [**2.3 Data Description**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3rdcrjn) | | | | | | | | | | | | | | | [**7**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3rdcrjn) | | |  |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |
|  | [**2.4 Import Data into Database**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.26in1rg) | | | | | | | | | | | | |  | | [**7**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.26in1rg) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | |  |  |  |  | |  |
|  | [**2.5 Export Data from Database**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.lnxbz9) | | | | | | | | | | | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.lnxbz9) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | | |  |  |  | |  |
|  | [**2.6 Data Pre-Processing**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.35nkun2) | | | | | | | | | |  | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.35nkun2) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | | | |  |  | |  |
|  | [**2.7 Modelling**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1ksv4uv) | | | | | | | | | |  | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1ksv4uv) | | | |
|  |  | | |  |  |  |  | | |  | |  | | | |  |  | |  |
|  | [**2.8 UI Integration**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.44sinio) | | | | |  | | | | | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.44sinio) | | | |
|  |  | | | |  |  |  | | |  | |  | | | |  |  | |  |
|  | [**2.9 Data From User**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.2jxsxqh) | | | | | |  | | | | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.2jxsxqh) | | | |
|  |  | | | |  | |  | | |  | |  | | | |  |  | |  |
|  | [**2.10 Data Validation**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.z337ya) | | | | | | | | |  | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.z337ya) | | |  |
|  |  | | | |  | |  | | |  | |  | | | |  |  | |  |
|  | [**2.11 Rendering the Results**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3j2qqm3) | | | | | | | | | | |  | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3j2qqm3) | | |  |
|  |  | | | |  | |  | | | | |  | | | |  |  | |  |
|  | [**3 Deployment**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1y810tw) | | | |  | | | | | | |  | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1y810tw) | |  |
|  | [**3.1 Unit Test Cases**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.4i7ojhp) | | | | | |  | | | | | | | | | [**9**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.4i7ojhp) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

3

Low Level Design (LLD)

**Abstract**

Our aim is to develop a Machine learning model and testing the model by using the data in relating to previous 6 months payment behaviour which is behavioural data and personal information which is demographic data as input of a client is used for this study.

     The research study is conducted using Random Forest Algorithm , Decision tree , Logistic Regression. Our aim is to identify that credit card customer is likely to default in the coming month.

     Credit risk plays a major role in the banking industry. Banking's main activities include granting loans, credit cards,investments, mortgages, etc. Credit cards are one of the fastest growing financial services offered by banks in recent years.

      However, as the number of credit card users increases, banks are facing rising credit card failure rates. Therefore,data analytics can provide solutions to address current phenomena and manage credit risk.

4

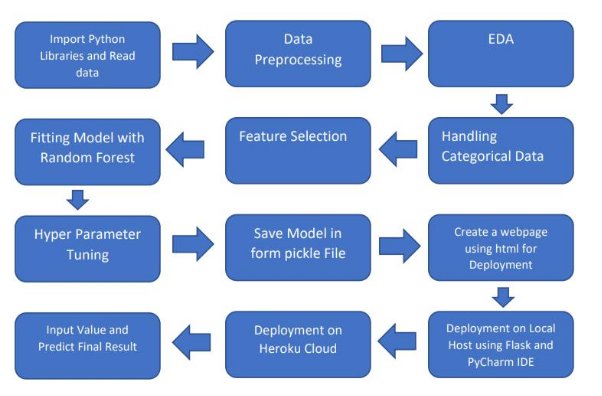
Low Level Design (LLD)

**1 Introduction**

**1.1 Why this Low-Level Design Document?**

The main purpose of this LLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

**1.2 Architecture**



**2. Architecture Design**

     Architecture is design in way that is start from importing the python libraries and load the data then process the data then perform the exploratory analysis .then feature scaling then train the model for prediction and predict the output.

5

Low Level Design (LLD)

**2.1. Data Gathering**

The data for the current project is being gathered from Kaggle dataset, the link to the data is: https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

**2.2. Tool Used**

• Python 3.7 is employed because the programming language and frame works like numpy, pandas, sklearn and alternative modules for building the model.

* • PyCharm is employed as IDE.

* • For visualizations seaborn and components of matplotlib are getting used

* • For information assortment prophetess info is getting used version

management.

• Netlify is employed for deployment

**2.3 Data Description**

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

 here are 25 variables:

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)

default.payment.next.month: Default payment (1=yes, 0=no)

6

Low Level Design (LLD)

**2.4 Import Data into Database**

* • Created associate api for the transfer of the info into the Cassandra info, steps performed are:

* • Connection is created with the info.

* • Created a info with name defaulter\_creditcard.

* • Cqlsh command is written for making the info table with needed parameters.

* • And finally, a cqlsh command is written for uploading the knowledgeset into data table by bulk insertion.

**2.5 Export Data into Database**

In the above created api, the download url is also being created, which downloads the data into a csv

file format.

**2.6 Data Preprocessing**

Steps performed in pre-processing are:

* • First the info sorts square measure being checked and located solely the value column is of sort number.

* • Checked for null values as there square measure few null values, those rows square measure born.

* • Converted all the desired column into the date time format.

* • Performed one-hot cryptography for the desired columns.

* • Scaling is performed for needed information.

* • And, the info is prepared for passing to the machine learning formula

7

Low Level Design (LLD)

**2.7 Modelling**

The pre-processed information is then envisioned and everywhere the specified insights are being drawn. Though from the drawn insights, the info is at random unfold however still modelling is performed with completely different machine learning algorithms to form positive we tend to cowl all the chances and eventually, for sure random forest regression performed well and any hyperparameter calibration is finished to extend the model’s accuracy.

**2.8 UI Integration**

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.Note CSS and HTML is not done by me.

**2.3 Data from User**

The data from the user is retrieved from the created HTML web page.

**2.4 Data Validation**

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

**2.11 Rendering Result**

The data sent for the prediction is then rendered to the web page.

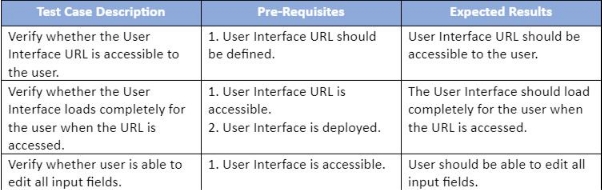
**3. Deployment**

The tested model is then deployed to aws using elastic benstalk. So, users can access the project from any internet devices.

8

Low Level Design (LLD)

**3.1 Unit Test**



9