Low Level Design

Credit Risk Predictor

Revision Number: 1.0
Last date of revision: 1/8/23

Pothuri Harish Varma

Document Version Control

Date	Version	Description	Author
01-08-2023	V1.0	LLD-V1.0	Pothuri Harish Varma

Contents

1. Introduction

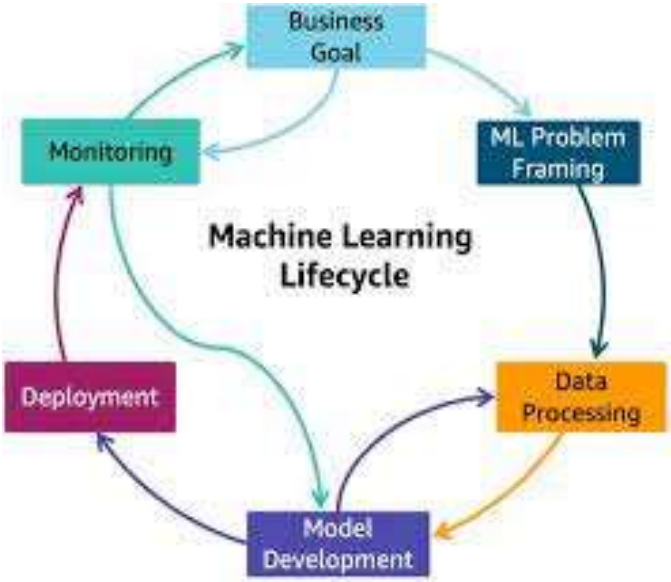
1.1. Why this Low Level Design Document?

The goal of LLD or Low-Level design document (LLDD) is to give the internal logical design of the actual program code. Low-Level design is created based on the High-Level design. LLD describes the class diagrams with the methods and relations between classes and programspecs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture



3. Architecture Description

3.1. Data Description

The primary source of data for this project is taken from UCI repository. The data contains 1000 records and 21 attributes. The data is in structured format and stored in ASC file

Import the ASC Data as Pandas DataFrame

```
In [2]: import pandas as pd
        df = pd.read_csv("C:/project1/notebook/data/SouthGermanCredit.asc", sep=' ')
        df
```

Out[2]:

	laufkont	laufzeit	moral	verw	hoehe	sparkont	beszeit	rate	famges	buerge	...	verm	alter	weitkred	wohn	bishkred	beruf	pers	telef	gastarb
0	1	18	4	2	1049	1	2	4	2	1	...	2	21	3	1	1	3	2	1	2
1	1	9	4	0	2799	1	3	2	3	1	...	1	36	3	1	2	3	1	1	2
2	2	12	2	9	841	2	4	2	2	1	...	1	23	3	1	1	2	2	1	2
3	1	12	4	0	2122	1	3	3	3	1	...	1	39	3	1	2	2	1	1	1
4	1	12	4	0	2171	1	3	4	3	1	...	2	38	1	2	2	2	2	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	1	24	2	3	1987	1	3	2	3	1	...	1	21	3	1	1	2	1	1	2
996	1	24	2	0	2303	1	5	4	3	2	...	1	45	3	2	1	3	2	1	2
997	4	21	4	0	12680	5	5	4	3	1	...	4	30	3	3	1	4	2	2	2

3.2. Preprocessing

In this stage data is prepared for analysis. This includes cleaning the data, removing duplicates, handling missing values, and transforming the data into a suitable format for machine learning models. In the input South German bank data there were no duplicate values or null values which needed to be handled.

3.3. Model Selection and Training

In this stage, a suitable machine learning model is selected based on the problem statement and data characteristics. The selected model is then trained on the preprocessed and engineered data to learn the patterns and relationships between the input and output variables. In this problem Random forest, Xgboost, Gradient Boosting, Adaboost and Bagging Classifier algorithms were hyper tuned to give the maximum accuracy.

3.4. Model Evaluation

The performance of the trained model is evaluated using various metrics such as accuracy, precision, recall, R2 score. This helps to determine the effectiveness of the model and identify areas for improvement.

3.5. Deployment and Monitoring

The final stage involves deploying the model into a production environment and continuously monitoring its performance. This includes tracking the model's accuracy and detecting any deviations from the expected behavior.

4. Unit Test Cases

Test Case Description	Pre - Requisites	Expected Results
Verify whether the Webpage is accessible to the User or not.	Webpage URL should be defined.	Webpage should be accessible to the User.
Verify whether the webpage is completely loaded for the User or not.	1. Webpage URL is accessible. 2. Webpage is deployed.	The Webpage should be completely loaded for the User when it is accessed.
Verify whether the user is able to enter data in input fields or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to enter data in input fields.
Verify whether the user is able to submit details or not.	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User is able to submit details to the process.
Verify whether the user gets recommended Results on submitting the details or not	1. Webpage URL is accessible. 2. Webpage is deployed. 3. Webpage input fields are editable.	The User gets recommended results on submitting the details.