Low Level Design (LLD)

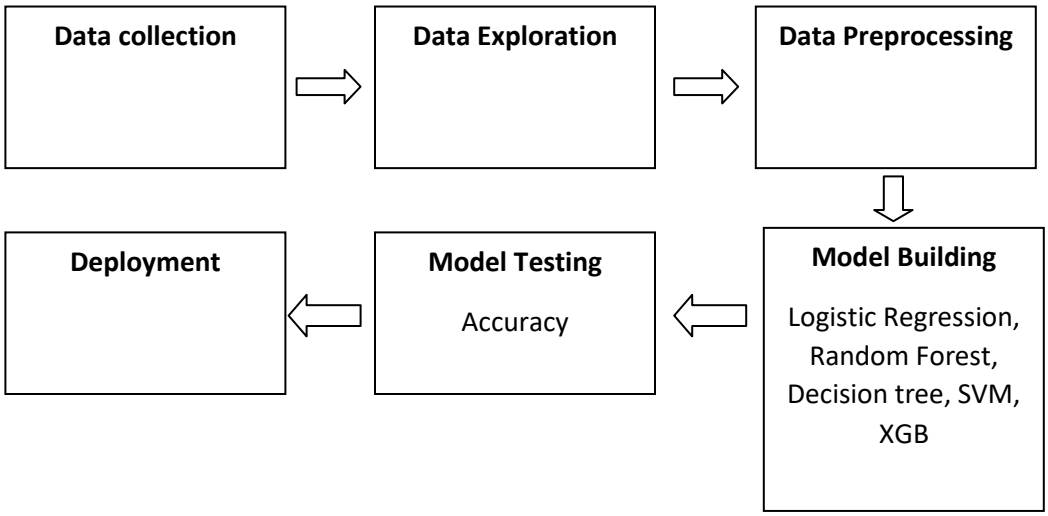
Credit Card Default Prediction

Table of contents

1. Introduction

The Low-Level-Design (LLD) gives an understanding of the internal logic design of the code for the defaulter prediction system. The LLD highlights different aspects of the architecture, the data involved and the various steps undertaken to reach the final goal of the project.

2. Architecture



3. Architecture Description

3.1 Data Description

The dataset used for the project the UCI Credit Card dataset which is in a .csv format consisting of 30000 rows of data relating to different credit card clients in Taiwan and 25 variables like demographic factors, history of payment, credit data and bill statements from April 2005 to September 2005

3.2 Import Data

Data is stored and imported to Python in CSV format which is then used for data pre-processing and model training and testing

3.3 Data Pre-processing

The dataset was fairly clean as it did not contain any null values and categorical variables Gender, Education etc were already encoded. Before building the models, the data was normalized using Min-Max scaler so that variables in the dataset are within a certain range. Feature scaling helps to keep the comparison between variables on common grounds. There was an imbalance in the dependent variable namely, the possibility of defaulting and this sorted by using the SMOTE method were samples in the minority class were synthetically oversampled to the level of the majority class to account for a fair classification.

3.4 Splitting the data

The dataset was split into train-test sets in 80:20 with the independent variables being the different features and the dependent variable being the possibility the client will default

3.5 Model Building

After cleaning and splitting the data, the machine learning models were built to understand which model best predicts the defaulter given a set of features. The algorithms built for the task were Logistic Regression, Random Forest, Decision tree, XG Boost and SVM which were then trained on the training dataset

3.6 Model Testing

The models were then tested on the testing dataset and the model that predicted the dependent variable with the highest accuracy was treated as the best model.

3.7 Deployment

The model will be deployed as an API using FastAPI where the user can input the relevant values to find if the customer is a credit card defaulter or not