CREDIT CARD DEFAULT CLASSIFICATION

Low Level Design (LLD)

Santhosh Kumar K S

NOVEMBER 2, 2023
INEURON

1. INTRODUCTION

1.1 What is Low Level Design Document?

Low-level design refers to the process of specifying and defining the detailed design of a software system. This Low level Design focuses on the implementation details of a system and is concerned with how the system will be built and how it will function at a detailed level. It provides the foundation for high-level design, which defines a system's overall architecture and design.

1.2 Scope of Low Level Design Document?

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.

2. PROJECT DESCRIPTION

2.1 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.

2.2 PROPOSED SOLUTION

Machine learning is a field in computer science aiming to imitate the human learning process. Machine Learning is a branch of Artificial Intelligence where computer learns from the data (past experiences) and makes future prediction. It finds the pattern in data, based on pattern it predicts for unseen data. Here we will develop machine learning models to predict the probability of credit default based on credit card owner's characteristics and payment history.

2.3 DATA INFORMATION

The dataset was taken from Kaggle (

URL: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset),
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.

Attribute Information:

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)

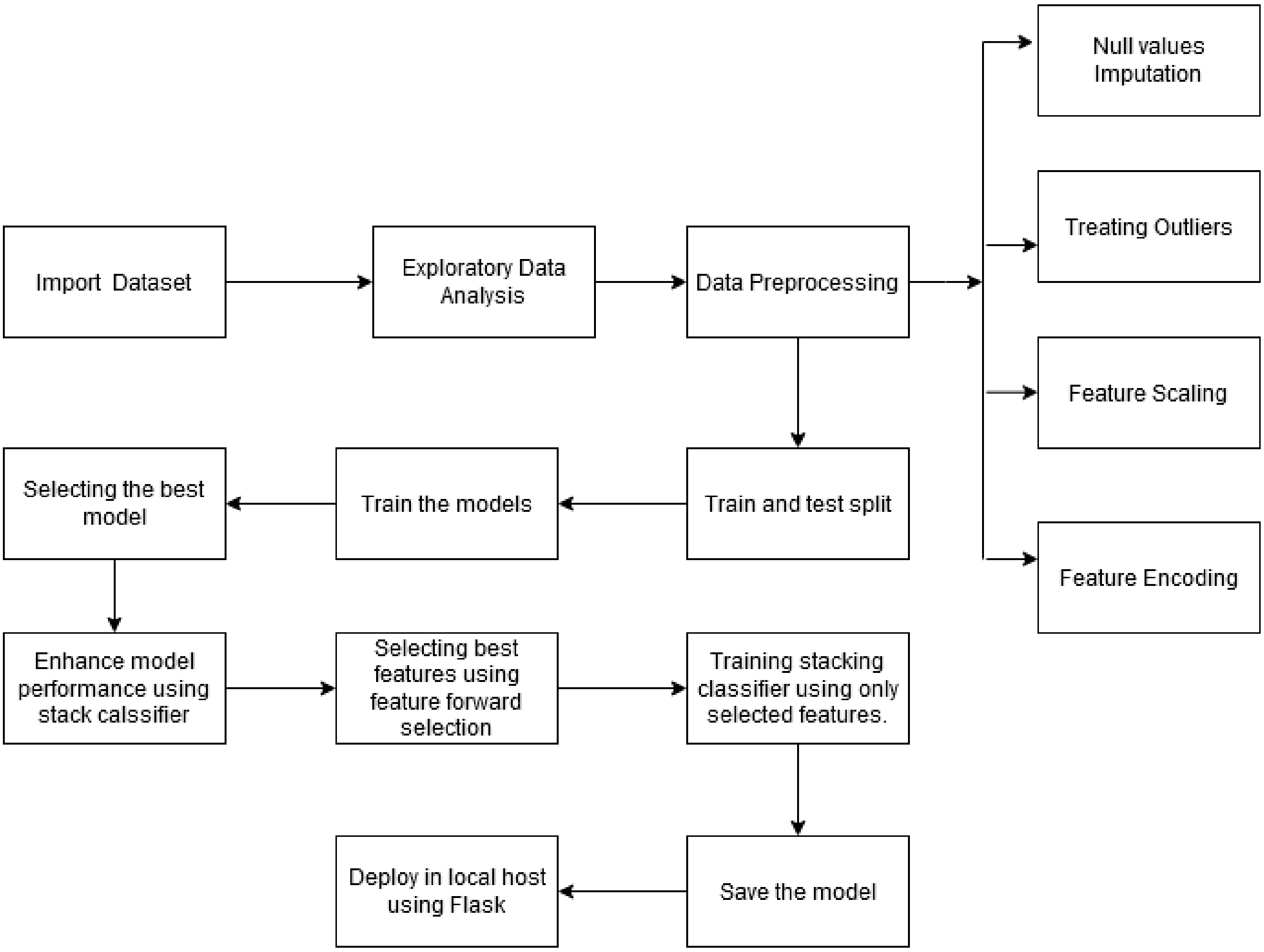
PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)

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)

3. ARCHITECTURE



4.Architecture Description.

4.1 Exploratory Data Analysis

In this exploratory data analysis we try to analyze each feature using visualization. Understanding the data distribution, understanding various facts that might leads model performance like null values, outlier detection etc.

4.2 Data Pre-processing.

In this step we will convert raw data into machine trainable format. In this we will go through various steps:

Null values Imputation:

This dataset has no null values.

Outliers Treatment:

In this dataset LIMIT_BAL, all 6 months PAY_AMT and 6 months of BILL_AMT having null values, so we replaced outliers with the median values.

Feature Scaling:

Some of features are ranged from -2 to 10 and some features are ranged from 0 to 1000000 so in this scenario we need to do feature scaling to make all features are in same range. And as these features are not normally distributed so we are using min-max scaling to scale the data.

Feature Encoding:

As these dataset has no categorical features so no need for feature encoding.

4.3 Train Test Split

This library was imported from Sklearn to divide the final dataset into the ratio of80-20%, where 80% of the data was used to train the model and the latter 20% was used to predict the same.

4.4 Train the Models

We tried and tested multiple models such as XGBoost, RandomForest,DecisionTree,Gradient Boost, CatBoost, Logistic Regression and KNearestNeighbour. XGBoost, Gradient Boost and CatBoost are giving best results.

Using these 3 models developed the stacking classifier and model performance improved.

4.5 Feature Engineering.

In this section we used Sklearn featureforward selection technique to select top k features which gives best result. Again trained with these features with stacking classifiers. And saved the model

4.6 Deploying using Flask

We designed UI and deployed model using Flask framework.

5. Screenshots