Low Level Design (LLD)

Credit Risk System

Revision Number: 1.0
Last date of revision: 22/12/2021

Jeevan Shriram Arande

Raj Chandeshwar Shukla

1. Introduction

Normally, most of the bank's wealth is obtained from providing credit loans so that a marketing bank must be able to reduce the risk of non-performing credit loans. The risk of providing loans can be minimized by studying patterns from existing lending data. One technique that you can use to solve this problem is to use data mining techniques. Data mining makes it possible to find hidden information from large data sets by way of classification.

The goal of this project, you have to build a model to predict whether the person, described by the attributes of the dataset, is a good (1) or a bad (0) credit risk

2. Problem Statement

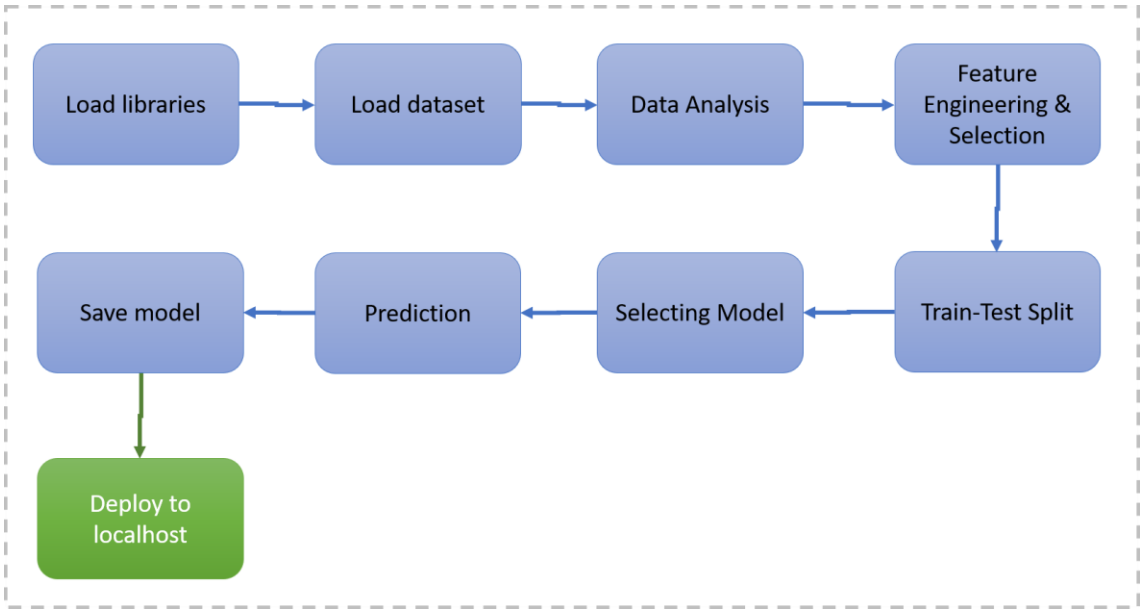
To create an ML solution for Credit Risk Prediction and to implement the following use case:

1. Based on the existing current customer risk data, build a model to predict a credit risk profile (Good or Bad).
2. Mitigate the risk by not providing loans to people with non-performing loans.
3. Provide loans to customers with good credit history.

3. Database Information

status: status of the debtor's checking account with the bank (categorical)
duration: credit duration in months (quantitative)
credit_history: history of compliance with previous or concurrent credit contracts (categorical)
purpose: purpose for which the credit is needed (categorical)
amount: credit amount in DM (quantitative; result of monotonic transformation; actual data and type of
savings: debtor's savings (categorical)
employment_duration: duration of debtor's employment with current employer (ordinal; discretized quantitative)
installment_rate: credit installments as a percentage of debtor's disposable income (ordinal; discretized quantitative)
personal_status_sex: combined information on sex and marital status; categorical; sex cannot be recovered from the
other_debtors: Is there another debtor or a guarantor for the credit? (categorical)
present_residence: length of time (in years) the debtor lives in the present residence (ordinal; discretized quantitative)
property: the debtor's most valuable property, i.e. the highest possible code is used. Code 2 is used, if codes 3
age: age in years (quantitative)
other_installment_plans: installment plans from providers other than the credit-giving bank (categorical)
housing: type of housing the debtor lives in (categorical)
number_credits: number of credits including the current one the debtor has (or had) at this bank (ordinal, discretized

job: quality of debtor's job (ordinal)
people_liable: number of persons who financially depend on the debtor (i.e., are entitled to maintenance) (binary,
telephone: Is there a telephone landline registered on the debtor's name? (binary; remember that the data are
foreign_worker: Is the debtor a foreign worker? (binary)



4. Architecture Description

4.1. Data Description

The widely used Statlog German credit data ([Web Link]), as of November 2019, suffers from severe errors in the coding information and does not come with any background information. The 'South German Credit' data provide a correction and some background information, based on the Open Data LMU (2010) representation of the same data and several other German language resources.

4.2. Data Pre-processing

This included importing of important libraries such as matplotlib, pandas, sklearn etc. We imported the same dataset mentioned above.

4.3. Data Analysis

Here we handled the null values, changed the column names, plotted multiple graphs in matplotlib and other visualization library for proper understanding of the data and the distribution of information in the same. As there were no null values in the data, we proceeded with the visualization and analysis.

4.4. Feature Engineering & Selection

Feature Selection using ANOVA for continuous variables and Chi Square Analysis for Categorical columns. We converted the nominal categorical columns using one-hot encoding and scaled the data using Min Max Scalar.

4.5. Train/Test Split

This library was imported from Sklearn to divide the final dataset into the ratio of 70-30%, where 70% of the data was used to train the model and the latter 30% was used to predict the same.

4.6. Selecting Model

We tried and tested multiple models such as XGBoost, RandomForest, Decision Tree, Logistic Regression and Naïve Bayes for the model and came up with the model with the best performance, i.e the Random Forest Classifier.

4.7. Prediction

The Accuracy of Random Forest was 61% and the F1 score was 69%.

4.8 Save Model

Model and encoders were saved using the pickle library which saves the file in a binary mode.

4.9 Deploy in Local Host

We created a HTML template and deployed the model through FastAPI.