Abstract : Home Credit Default Risk Kaggle Competition

Introduction :
Loans have always been an important part of people's lives. Each individual has different reasons for borrowing a loan. It could be to buy a dream car or a home, to set up a business, or to buy some products.
A large part of the population finds it difficult to get their home loans approved due to insufficient or absent credit history. It is a major challenge for banks and other finance lending agencies to decide for which candidates to approve housing loans.
A machine learning-based classification tool to predict loan default risk which uses more features than just the traditional credit history can be of great help for both, potential borrowers, and the lending institutions.

Problem Statement : The problem can be described as, "A binary classification problem where the inputs are various features describing the financial and behavioral history of the loan applicants, in order to predict whether the loan will be repaid or defaulted."

Data:
The dataset received is from the Home Credit Group as it is a clean set.
Home Credit is an international consumer finance provider that provides point of sales loans, cash loans, and revolving loans to underserved borrowers.The dataset files are provided on the Kaggle website in the form of multiple CSV files and are free to download

Project Design and Solution

The project has been divided into five parts-

1. Data Preparation - Before starting the modeling, we need to import the necessary libraries and the datasets. If there are more than one file, then all need to be imported before we can look at the feature types and number of rows/columns in each file.
2. Exploratory Data Analysis - After data importing, we can investigate the data and answer questions like- How many features are present and how are they interlinked? What is the data quality, are there missing values? What are the different data types, are there many categorical features? Is the data imbalanced? And most importantly, are there any obvious patterns between the predictor and response features?
3. Feature Engineering - After exploring the data distributions, we can conduct feature engineering to prepare the data for model training. This includes operations like replacing outliers, imputing missing values, one-hot encoding

categorical variables, and rescaling the data. Since there are a number of relational databases, we can use extract, transform, load (ETL) processes using automated feature Engineering with Feature Tools to connect the datasets. The additional features from these datasets will help improve the results over the base case (logistic regression).

4. Classifier Models: Training, Prediction and Comparison - After the dataset is split into training and testing sets, we can correct the data imbalances by undersampling the majority class. Then, we can training the different classifier models (Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes, XGBoost, Gradient Boosting, LightGBM) and compare their performance on the test data using metrics like accuracy, F1-score and ROC AUC. After choosing the best classifier, we can use K-fold cross validation to select the best model. This will help us choose parameters that correspond to the best performance without creating a separate validation dataset.
5. Hyperparameter Tuning - After choosing the binary classifier, we can tune the hyperparameters for improving the model results through grid search, random search, and Bayesian optimization (Hypertext library). The hyperparameter tuning process will use an objective function on the given domain space, and an optimization algorithm to give the results. The ROC AUC validation scores from all three methods for different iterations can be compared to see trends.