Abstract : Home Credit Default Risk Kaggle Competition

The Home Credit Group needs a machine learning-based classification model to predict loan default risk which uses features beyond just the traditional credit history and can make accurate lending decisions for individuals. The data consists of eight relational tables, which contain applicants' static data such as their gender, age, number of family members, occupation, and other related fields, and applicant's previous credit history. We conduct feature engineering to prepare the data for model training with operations like replacing outliers, imputing missing values, one-hot encoding categorical variables, and rescaling the data. After the dataset is split into training and testing sets, we can correct the data imbalances by undersampling the majority class. Then we can train the different classifier models using Logistic Regression or Gaussian Naive Bayes and compare their performance on the test data using metrics like accuracy, F1-score, and ROC AUC. After choosing the best classifier, we use K-fold cross-validation to select the best model. The hyperparameter tuning process will use an objective function on the given domain space, and an optimization algorithm to give the results. Our study predicts that the baseline model with all features has the test AUC score in notebook(0.7434) , we expect to improve on this by doing more feature engineering like performing dimensionality reduction, SVC etc.

Project Execution Plan

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.
The file application_{train|test}.csv contains the main table containing the training dataset and test dataset, with each row representing one loan identified by the feature SK_ID_CURR. The training set contains the variable TARGET with binary values (0: the loan was repaid or 1: the loan was not repaid). There are many input files available, which will be analyzed for input features to train the model. The large number of input features and training samples will allow me to identify the important factors and for constructing a credit default risk classification model.

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.