RICKY SU

Competition Description: (Home Credit Default Risk)

Due to insufficient and non-existing credit histories, many people struggle to get loans. Unfortunately, this community is often taken advantage of by lenders. In order to make sure this underserved population has a positive loan experience, Home Credit, an international finance consumer provider, provided datasets that challenge Kagglers unlock the potential dataset to predict their clients' repayment abilities. For the Home Credit Default Risk competition, the performance matrix used is the Area Under the ROC Curve (AUC-ROC). It enables financial institutions to extend credit to underserved populations responsibly.

Dataset:

The dataset provided by Home Credit consists of multiple tables, each containing different types of information related to loan applications and client credit histories.

- application_{train|test}.csv
  This is the main table containing static data for all applications with one row representing one loan. The target variable, TARGET, indicates whether the client has repayment difficulties (1 for difficulties, 0 for no difficulties).
- bureau.csv
  This table records previous credits from other institutions reported to the Credit Bureau.
- bureau_balance.csv
  This contains monthly balances of previous credits in the Credit Bureau.
- POS_CASH_balance.csv
  Monthly balance snapshots of previous point-of-sale (POS) and cash loans with Home Credit.
- credit_card_balance.csv
  Monthly balance snapshots of previous credit cards with Home Credit.
- previous_application.csv
  All previous applications for Home Credit loan.
- installments_payments.csv
  Repayment history for previously disbursed credits.
- HomeCredit_columns_description.csv
  Descriptions of columns in the various data files

ML Problem Formulation:

The problem can be transformed into a **binary classification** problem which predicts whether a client will have repayment difficulties (TARGET=1) or not (TARGET=0). We can evaluate the model performance using the receiver operating characteristic (**ROC**) **curve** which is the plot of the true positive rate (TPR) against the false positive rate (FPR) at each thresholding setting. It measures the model's ability to distinguish between the two classes.

1. **Data Preprocessing**: The goal of data preprocessing is to prepare the raw data for machine learning by cleaning and aggregating tables.
    a. Aggregate related tables: Combine related tables together
    b. Handling missing values: Identifying the missing values and drop columns with a high percentage of missing values.
    c. Encoding categorical Variables
    d. Feature Engineering:
2. **Feature Selection**: The goal of feature selection is to identify the most important features that have the highest impact on predicting the target variable (TARGET).
    a. Correlation Analysis: Calculate the correlation between each feature and the target variable (TARGET)
    b. Dimensionality Reduction: If the dataset has too many features, try to select a certain amount of features to reduce dimensionality
3. **Model Development**: The goal is to train multiple models and evaluate their performance using the ROC curve.
    a. Model Selection: Train on at least three different models (ML algorithms)
    b. Model Evaluation: Split the training data into training set and validation set (80% training and 20% validating) use the AUC-ROC curve to evaluate the model performance.
    c. Model Prediction: Use the best performance model to predict the target variable (TARGET) on the testing data.

ML Algorithms:

Logistic Regression:
Logistic regression is a supervised linear machine learning algorithm that accomplishes binary classification problems by predicting the probability of an outcome. The advantages of using logistic regression are that it can work with both numerical and categorical features and assess what variables are useful for classifying samples.It predicts probability directly using a sigmoid function.

Random Forests:
Random forest algorithm is a supervised nonlinear classification model. It contains two critical concepts — bootstrap and feature selection. It predicts the class by averaging probabilities from multiple independent decision trees.

XGBoost:
XGBoost is an ensemble learning method that combines multiple decision trees to create a stronger predictive model. It works by building trees sequentially, where each new tree corrects the errors made by the previous trees. XGBoost has native mechanism to handle missing values in tabular data. It automatically ranks feature variables by their predictive power and handles

mixed data types effectively. It sums up the contributions from sequential built trees and makes the classification.

Action Plan:

1. Data Exploration and Understanding (March 24 - 30) - Understanding the dataset
   Explore all tables in the dataset thoroughly
   Analyze relationships and dependencies between tables
2. Data Preprocessing (March 31 - April 8) - Clean the raw tables and datasets
   Clean and prepare each table
   Handle missing values
   Encode categorical feature variables
   Standardization/ Normalization on numeric features
3. Feature Engineering (April 9 - 15) - Prepare the feature set for ML training
   Creating new features by aggregating, calculating ratios and interactions
   Merging related tables
   Domain-specific feature creation
4. Model Development (April 16 - 27) - Train all three models on this same feature set
   Implement Logistic Regression (may need to standardize features)
   Build Random Forest model
   Develop XGBoost model
   Split the training data into training set and validating set (80% - 20%)
   Plot the three ROC curves in the same graph to evaluate the models' performance
5. Final Report (April 27 - May 4) - Complete the final report
   Generate the prediction on the testing dataset
   Write the final report

Potential Pitfalls and Alternative Approaches:

This project faces several potential challenges. Class imbalance in credit default data may affect model performance. Time constraints are a significant concern. If I run out of time, I'll prioritize the most important features from bureau and previous_application tables before exploring complex interactions. I also set the deadline of my project 2 days prior to the final deadline as a buffer. If the selected algorithms underperform, I'll pivot to alternatives like a simple decision tree or other ML model for binary classification problems.