Home Credit Default Risk

Risk analysis of Home credit clients using machine learning

Problem statement

Intro

- The dataset provided by Home Credit on kaggles lists credit loans historical data for their clients
- The dataset contains multiple files such as:
  - application_{train/test}: This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET). Static data for all applications. One row represents one loan in our data sample.
  - Bureau: All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in our sample).
  - etc..

Exploratory data analysis

I conducted exploratory data analysis to describe the main dataset and obtain initial descriptive statistics for it

The main finding is that the data is highly imbalanced, i.e. only 8% of the data is for defaulted loans while the rest are loans paid without any difficulty

This is a challenge for machine learning algorithms as the model must be carefully calibrated to accurately model the minority class

As in our case we are much more interested in finding what criteria contributes to a loan default more than a successful loan, as it is much more costly to approve a potential risky client than it is to decline a potentially reliable client

Algorithm choice

I decided to use **XGBoost** as the main model as it is a highly effective and efficient ML model for tabular data, it natively handles missing data which greatly reduces imputation efforts and prevents against disrupting the original distribution of the data by the imputation methods.

It also (as with ensemble tree models) can capture complex interdependencies and relationships between variables that otherwise would be difficult for linear models

Model training

I trained the model locally on CPU using a set of hyperparameters which i found on kaggle kernels due to the time limitations as I couldn't perform cross validation as it is a time and computation intensive process.

Using the base model I was able to obtain a 76% validation AUC score which is decent without any attempts at feature engineering or model tuning.

Unfortunately the model suffered from relatively high false positive rate which is detrimental to the original task.

Feature Engineering

It is critical to utilize features provided by the other datasets in order to enhance our model performance.

Duel to my lack of domain knowledge in this problem I decided to use simple aggregations of client previous scores with some best practice tips from kaggle experts who provided excellent insights for the additional features

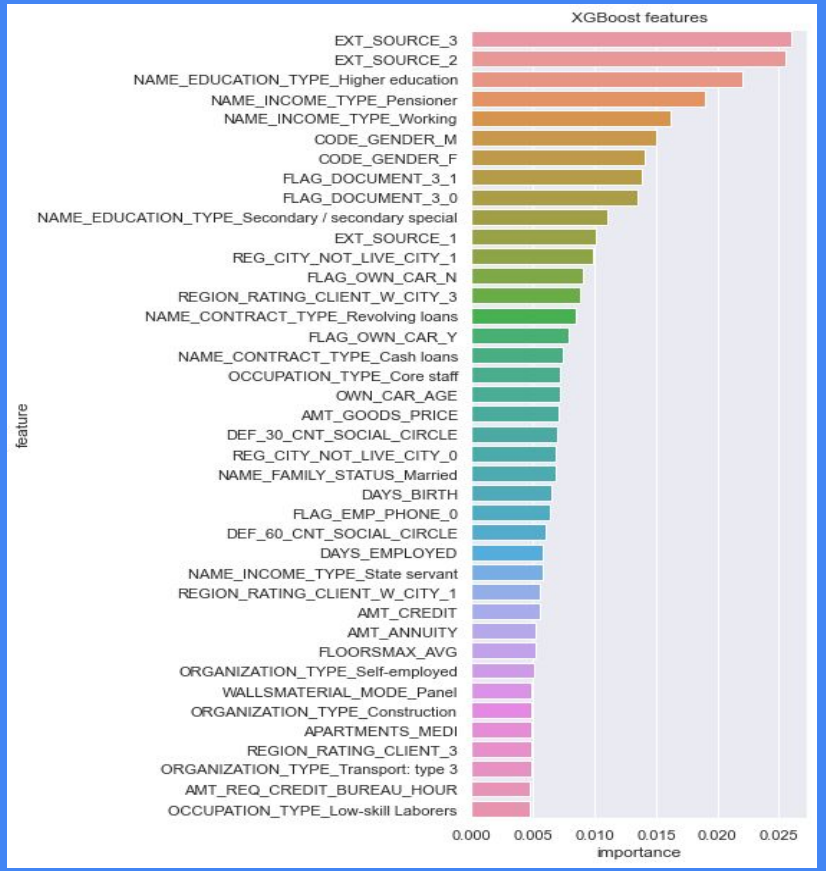
Next i trained the model with all the extra features and obtained 7x% Validation AUC score which is a significant improvement

Feature importance

A key feature of tree based models is that they provide a relative importance for features used in prediction which can guide decision making and help with model explainbilty

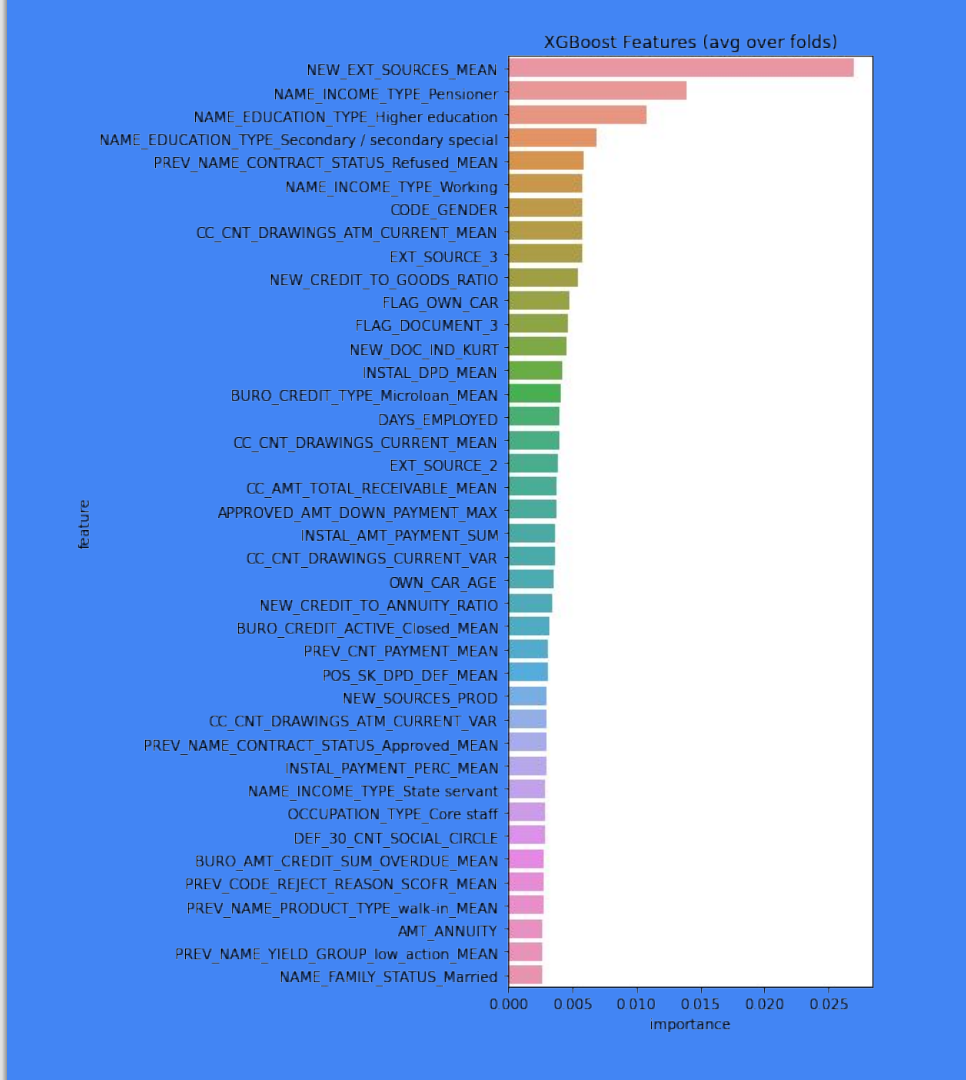
Feature importance

Base model



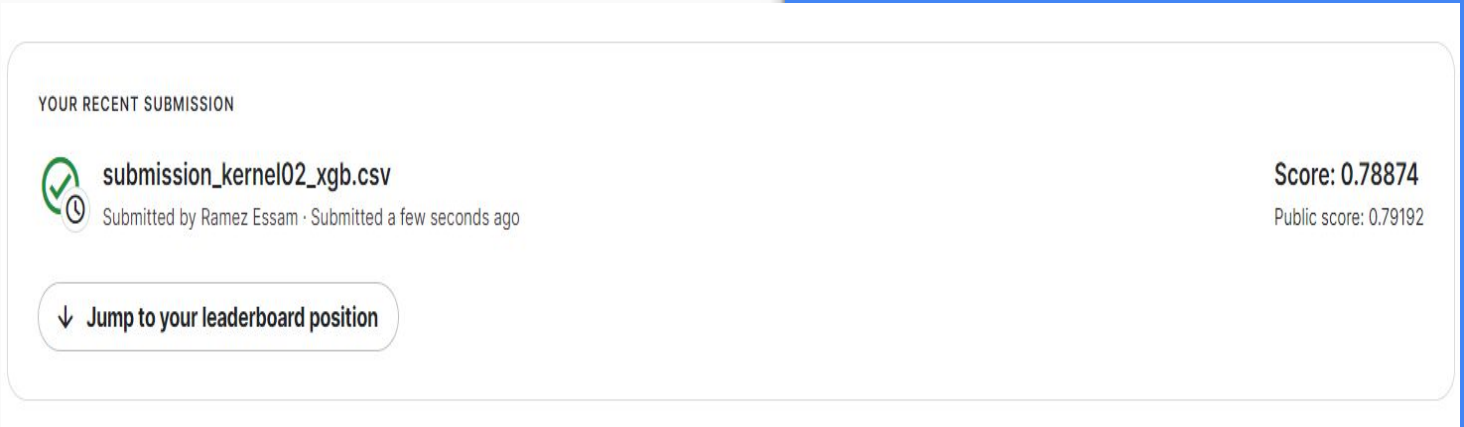
Feature importance

Enhanced model using feature engineering



Final Submission

Using the enhanced model



The most important takeaway is that while choice of model and hyperparameters is crucial, it eventually comes down to extracting useful features from the data for the model to learn from

Thanks for your time