Credit Risk Analysis Using LightGBM

Henry Ker Christian Ibanez-Diaz Braedon Shick

University of Michigan

Project Statement

For our capstone project, we chose a Kaggle competition[1] sponsored by Home Credit, a housing loan provider. The objective of this contest is to develop a model that predicts whether a client will default on a loan. This contest emphasizes the trade-off between short term performance and stability over time, and penalizes submissions which show deteriorating performance in later weeks. The creators provide economic and behavioral data, in the form of credit application history, credit bureau reports and other personal data.

This is the first Kaggle competition for any member of this group, so we see it as a chance to learn more about the platform and the competition dynamic. We chose this project for the opportunity it provided to work with a massive industry dataset, and the benchmark provided by other teams' submissions.

Project Goal

The competition scores submissions using a custom metric, derived from the gini coefficient of the ROC-AUC. The metric penalizes week to week variability, and heavily penalizes decreasing performance over time.

$$stabilityMetric = mean(gini) + 88 * min(0, bestfitslope) - 0.5 * std(bestfitresiduals)$$

A successful outcome of our capstone project would be ranking in the top 10% of all teams on the Kaggle leaderboard at the time of project submission (the contest continues for another month).

Data Description

The dataset comprises multiple entries spread across 17 categories of tables (sometimes subdivided further by timestamp). Each relation has from 3 to over a hundred features, and can represent either a single client metric or a client history which will need to be aggregated into a metric. In addition the dataset is heavily unbalanced.

The data was treated as follows in order methods to reduce the space that takes in memory and to reduce data dimensionality.

1. **Standardization of datatypes**.
2. **Datetime stamps conversion to float**. So that datetime variables could be quantify and evaluated in the model.
3. **Data Aggregation**. If multiple income information was available the maximum income was used.
4. **Feature Selection.** Using Shapley Additive explanations techniques (SHAP) values
5. **Missing Values Treatment**. Features were removed if less than 90% complete

Methodology

1. We utilized Polars for data loading and initial processing. Not only is Polars faster than pandas, but the lazy query evaluation allowed us to work with this dataset that would otherwise be too large to fit in local memory.

2. For feature selection, SHAP analysis helped up determine the importance of each feature in a model to the predicted result. This method was applied separately to each file in the dataset, and the top 10 most important features for each were recorded.

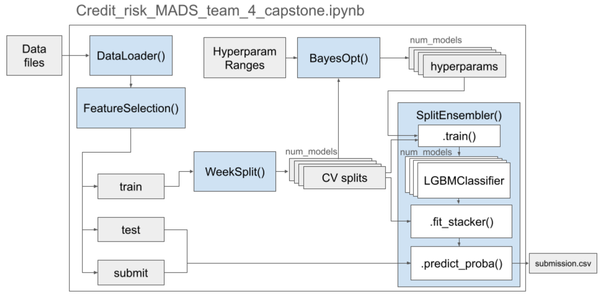
3. Before any processing was applied, we split off the last 30 weeks as a test set. We then randomly sampled 6 subsets from the train df, and further divided those into CV splits (maintaining week ordering to avoid data leakage). Each subset became the basis for a different model.

4. We created an ensemble of 6 lightgbm classifiers. Light GBM is a gradient boosting decision tree ensemble developed by researchers from Microsoft. Gradient boosting decision trees like LightGBM and XGBoost have been the state-of-the-art classification architecture for years now, and are often at the core of winning Kaggle classification models [2].

5. The model predictions were combined using a simple average (no performance increase was observed using more sophisticated stacking algorithms)

6. We leveraged Bayesian Optimization for hyperparameter tuning, controlled by a Weights and Biases sweep. Bayesian Optimization starts with random choices from the hyperparameter space, and iteratively refines its predictions using a balance of exploration and exploitation.

Code Structure



Results

Our best performing ensemble achieved 0.84 ROC-AUC on our hold-out test data. The results by week for each model in the ensemble are shown below, with out-of-sample weeks represented by a line and in-sample weeks for each model represented by points.

Discussion

Despite achieving a competition metric score of 0.645 on our test data, our best performing submission scored only 0.494: well below the top leaderboard score of 0.600, and in fact just below the 50th percentile score of 0.509. We attribute this to our model failing to generalize across a covariate shift in the hidden submission data. We had hoped that training models on smaller subsets of the data might allow them to specialize at predicting for different input distributions, but our architecture was not able to induce sufficient specialization (see similar patterns in above figure).

Next Steps

We expect that improvement could be achieved by modifying ensemble architecture to induce more specialization. Winning Kaggle models can often have quite complex architectures; ours is quite rudimentary in comparison to winning solutions such as [3] for a similar classification competition. Future modifications should include experimentation with more model types (e.g. neural networks and SVCs) and feature engineering informed by credit risk domain research.

References

[1] Home Credit Group. (2024). Home Credit - Credit Risk Model Stability. Kaggle.

[2] Kumar, S. (2024) Winning solutions of kaggle competitions. Kaggle.

[3] Tunguz, B. et. al. (2018) Home Credit Default Risk - 1st Place Solution. Kaggle.