MMF2000H: Risk Management - AML/ATF Risk

Ian (Min Jae) Lee, Joshua (Ha Rim) Kim,
Linlong (Liam) Wu, Zixun Zhai

November 25, 2022

Model

For this assignment, we adopted AutoML technique. The pool of machine learning algorithms from which *H2O* AutoML selected are:

- generalized linear model with regularization (GLM)
- distributed random forest (DRF)
- extremely randomized trees (XRT)
- gradient boosting machines (GBM)
- extreme gradient boosting (XGBoost)

The final model is a gradient boosting machine with the following hyperparameters:

- $number\_of\_trees = 31$
- $number\_of\_internal\_trees = 31$
- $max\_depth = 4$
- $min\_leaves = 6$

We measured ROC AUC on the test set and summarized these performance metrics as follows:

Table 1: *final model performance evaluation on the test set*

ROC AUC	accuracy	precision	recall	F_1-score
0.852	0.754	0.593	0.875	0.707

Question 1

We detected two identical "tot_acct_num" columns in the provided dataset and removed one of the duplicates. Once removed, the modified dataset, including both the features and the target, was clean and did not require any data transformation. We have verified the following from the dataset:

- there are no missing (e.g. *NaN* or empty elements) or faulty (e.g. string for a numerical feature) values;
- there are no duplicate data points, i.e. no two rows have identical values across all columns;
- there are no extreme outliers which deviate significantly from the general distributions of the features;
- each feature has a mean of 0 and a standard deviation of 1, indicating that the variables have already been standardized; and
- each categorical (binary indicator) feature has been numerically encoded with zeroes and ones, then also standardized.

Question 2

In machine learning, feature selection can be a crucial step if:

- the number of features is too large relative to the number of data points, which can lead to overfitted models;
- there are irrelevant or noisy variables which can also result in overfitting;
- time and computation resources are constrained – fewer features generally reduces training time and cost; or
- model interpretability is desirable so it is advantageous to identify important variables.

For this project, feature selection is necessary as the sample-to-feature ratio is low (588 samples vs. 149 features) and the explainability of the model is required in the context of AML business. Furthermore, removal of irrelevant features and faster training speed are beneficial when building the risk rating classifier.

The techniques used to perform feature selection are as follows:

1. Using the training set (which is further described in *Question 3* regarding the dataset split), build a simple random forest classifier where class weights are adjusted inversely proportional to class frequencies in the training set in order to address imbalanced class distribution.
2. Compute (Gini) feature importance values with the trained random forest classifier then sort the variables based on feature importance in descending order.

3. Plot the sorted feature importance values then employ the *elbow method* to determine the optimal number of features. An *elbow* is a cutoff point where diminishing returns are no longer worth the additional cost. In feature selection, one should choose a number of features so that adding another variable to the model does not give much better performance. This can be visualized on the plot as a point of inflection of the curve or an "elbow".

4. Select all features up to the elbow from the sorted list of features.

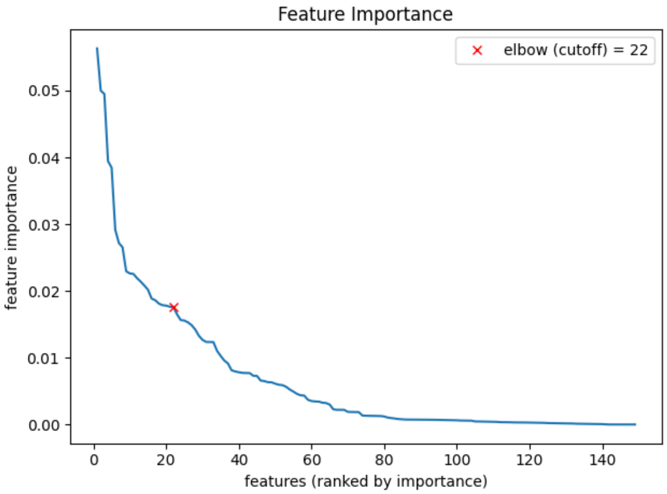

Figure 1: *plot of feature importance values of 149 variables and the elbow at 22*

Figure 1 illustrates how the elbow method has been applied to our dataset. We chose 22 as the elbow given the sharp corner and a noticeable drop in feature importance. Note that this technique does not pinpoint a unique elbow and does not provide an unambiguous cutoff. Hence selecting an elbow is a subjective process, making it one of the challenges further discussed in *Question 7*.

Question 3

The purpose of train-test split is to obtain an unbiased evaluation of a trained model with a dataset that was unused during training. If the training set were to be used to measure model performance, the evaluation would be overly optimistic since the model has already seen the training set before. For the project, we chose 80:20 split (80% of the data for training, 20% for testing), a commonly used split ratio in practice.

Sample weights were also taken into account when splitting because there is a discrepancy in the number of data points between two classes (198 high risk vs.

390 non-high risk targets). To achieve the unbiased nature of the test set and represent this class imbalance equally during the split, we assigned the same ratio of two target classes to both training and test sets.

Question 4

The model selection process was facilitated by *H2O*, a Python library offering an open-source automated machine learning framework, i.e. *AutoML*. AutoML is a process of automating various stages of machine learning from data preparation to model deployment. Since the project boils down to a classic binary classification problem using tabular data, a heavily studied topic in the field of supervised learning, such task is where AutoML excels. The pool of machine learning algorithms from which *H2O* AutoML selected are:

- generalized linear model with regularization (GLM)
- distributed random forest (DRF)
- extremely randomized trees (XRT)
- gradient boosting machines (GBM)
- extreme gradient boosting (XGBoost)

While the library also provides deep neural network and stacked ensemble, they were manually excluded from the pool because the two models lack interpretability compared to the other linear or tree-based algorithms. A convenient feature of *H2O* AutoML is the hyperparameter optimization using random grid search. The library documentation lists the hyperparameters of each base model, along with all potential values that can be randomly chosen in the search.

We ran the experiment for 50 model iterations, followed by the model selection stage where the AutoML framework chooses the model with the lowest 5-fold cross-validation log-loss (binary cross-entropy loss) as the final classifier. The final model is a gradient boosting machine (GBM) with the following hyperparameters:

- $number\_of\_trees = 31$
- $number\_of\_internal\_trees = 31$
- $max\_depth = 4$
- $min\_leaves = 6$

The classification threshold is 0.223020, which was calibrated such that it maximizes the F_1-score on the training set. The main objective of an AML risk classifier is to minimize the number of false negatives as missing even a single positive case can be detrimental to an organization. However, we should not aim to maximize the recall of the model because one can simply achieve 100% recall by labelling everything as positives. The F_1-score, calculated as the harmonic mean of the precision and recall, resolves this issue by striking the balance between the two metrics.

Question 5

To evaluate the model, we measured ROC AUC (area under the ROC curve) on the test set. ROC AUC assesses how well a binary classification model is able to distinguish between true and false positives. An AUC of 1 indicates a perfect classifier, while an AUC of .5 indicates a poor classifier, whose performance is no better than random guessing. The AUC of 0.852 suggests the our model is reasonably capable of classifying between high and non-high risk customers.

Summarized in *Table 2*, the four metrics - accuracy, precision, recall, and F_1-score - are measured to evaluate the quality of our model predictions, where they range between 0 (poor) and 1 (perfect). While all of them yield values greater than 0.5, demonstrating decent predictive capacity of the classifier, the high recall of 0.875 is especially promising since an ideal AML risk model should always be able to detect risky (positive) clients.

Table 2: *final model performance evaluation on the test set*

ROC AUC	accuracy	precision	recall	F_1-score
0.852	0.754	0.593	0.875	0.707

Question 6

Figure 2 illustrates the relative feature importance of all 22 variables used to build the classifier. To highlight some key features, “in_person_visit_cnt”, “tot_cross_border_val_12m”, and “tot_cross_border_cnt_12m” form the top 3 list, where they all have relative importance of 0.5 or greater. On the other hand, the feature that is deemed the least essential for the classifier is “tot_cash_val_5m”.

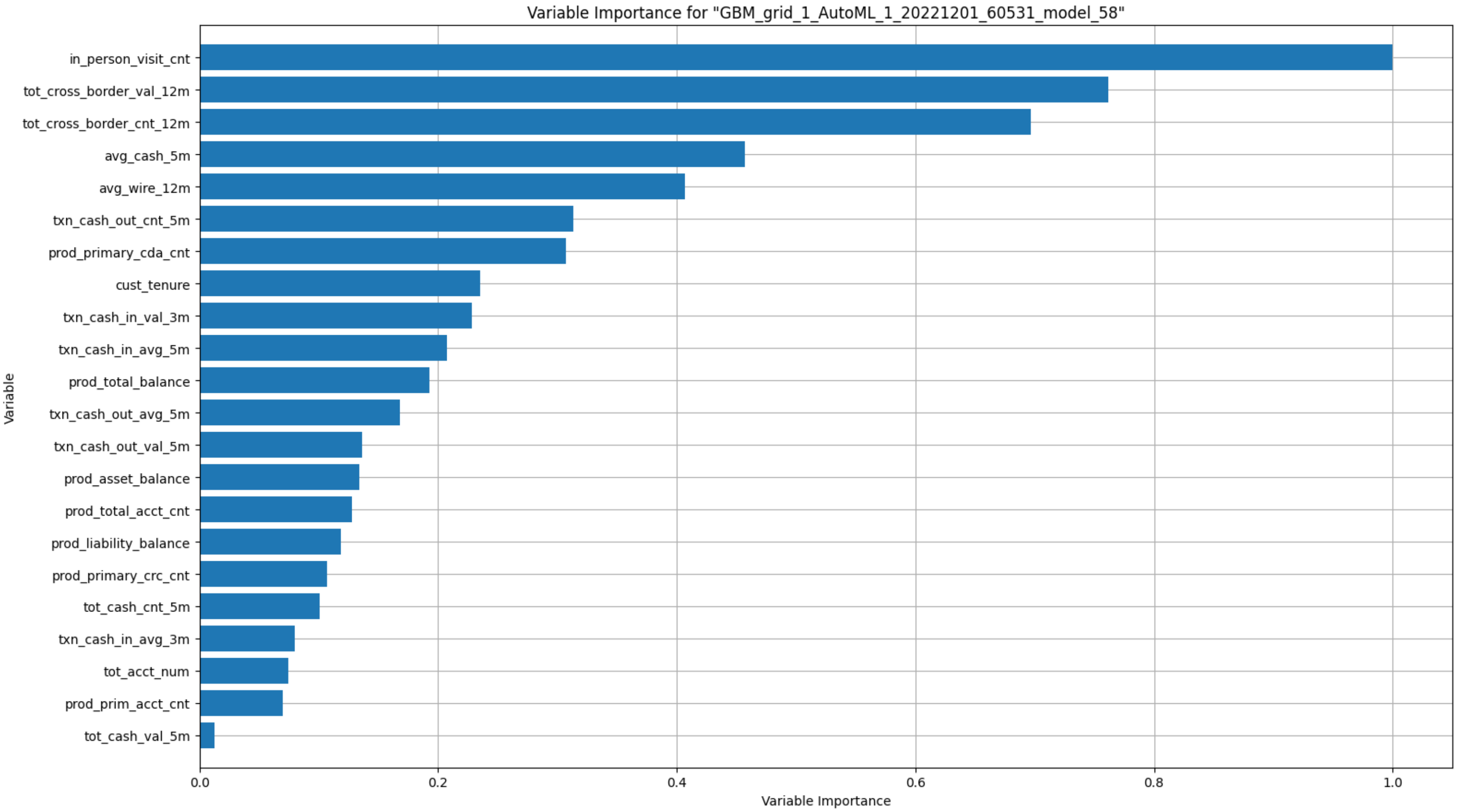

Figure 2: *bar chart of relative feature importance for the final AML risk classifier*

Figure 3 visually summarizes the *SHAP* values of the features evaluated on the test set. SHAP (**SH**apley **A**dditive ex**P**lanations) is a concept based on cooperative game theory and provides a principled way to explain the predictions of nonlinear models in the field of machine learning. SHAP value represents the individual contribution of each feature on the output of the model for each observation. The feature "in_person_visit_cnt", for example, displays the majority of positive SHAP values as shades of red and negative SHAP values as blue. This implies that the model generally associates "in_person_visit_cnt" and the target variable "rating" with a positive correlation.

Many features behave in a manner that aligns with our intuition, such as "cust_tenure" having negative impact on AML risk. Variables describing some form of average or total financial values (e.g. "avg_cash_5m", "txn_cash_in_avg_5m", and "txn_cash_out_val_5m") should intuitively have positive contribution to the risk target since the monetary amount involved in money laundering crimes would usually be greater than transactions carried out by ordinary customers by orders of magnitude. Similarly, features regarding cross-border transactions are also strong positive factors of AML risk.

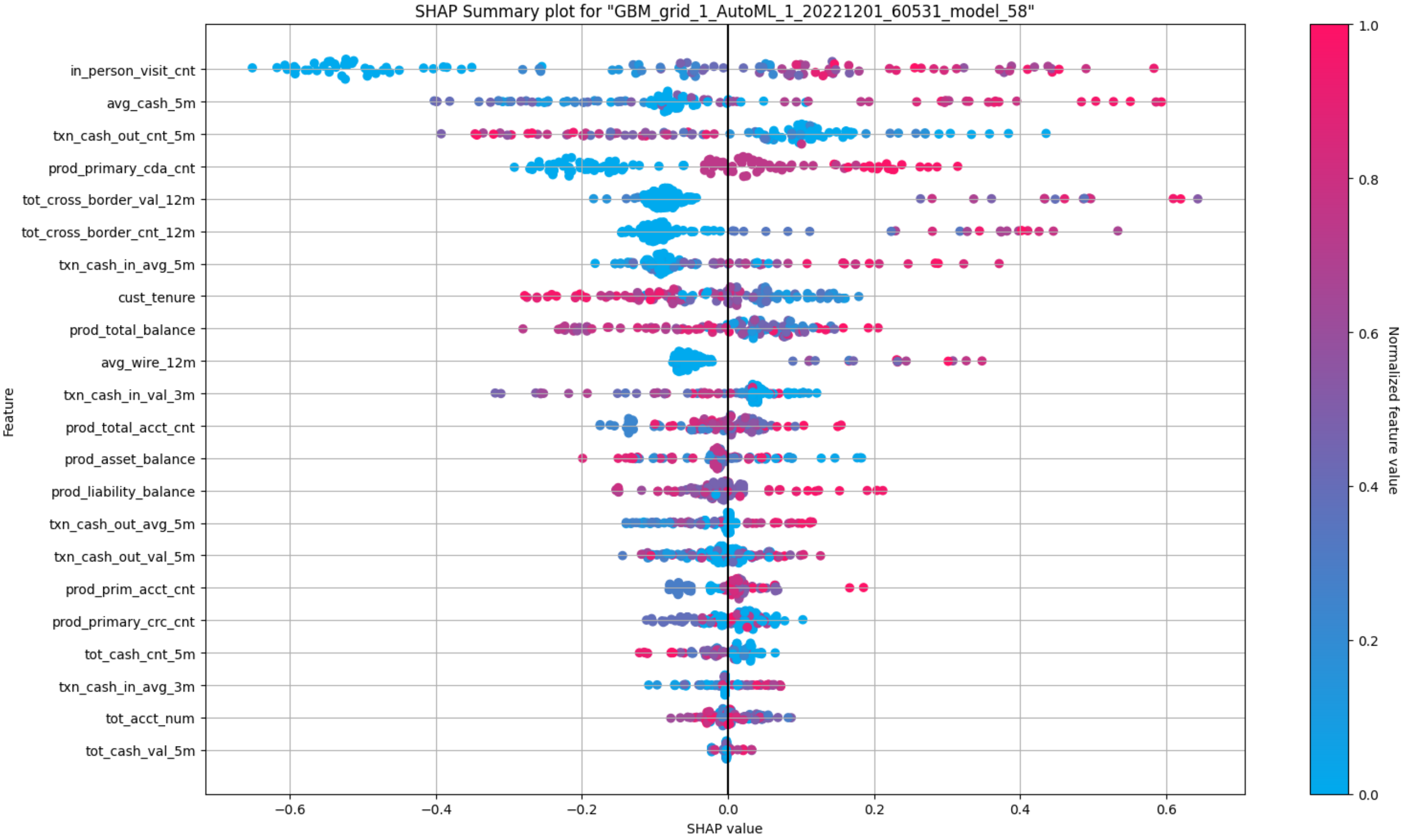

Figure 3: *SHAP summary plot for the final AML risk classifier evaluated on the test set*

At the same time, there exists some variables that portray counter-intuitive behaviours. Most notably, “in_person_visit_cnt” having positive correlation with the risk level is against our belief that money laundering criminals would refrain themselves from visiting bank branches in-person. Moreover, one would expect the criminals to be opposed to having accounts as primary holders to avoid surveillance from the bank as much as possible, which disagrees with the positive relationship between “prod_primary_cda_cnt” and the target variable.

The problem might arise from some significant features which are correlated with these counter-intuitive features have not been identified. These hidden features are correlated with default events. For example, an increase in "in_person_visit_cnt" may contribute to an increase in the number of trades or number of deposits under that particular customer. To address this, we might need to define more features and collect the corresponding data.

Question 7

The size of the dataset is relatively small, even after the feature selection procedure to accommodate for low sample-to-feature ratio, which can lead to several issues. Small training set often results in overfitting especially when algorithms, such as XGBoost or GBM, have high complexity. This is evident in our project from the discrepancy between the training AUC of 0.905 and the test AUC of 0.852. On the other hand, small test set may not represent the training set well enough, so the training performance does not translate to test evaluation metrics. To resolve them, one may collect more data points to increase the dataset size, improving the robustness of the training set and reducing the gap between the training and test set.

Furthermore, the *elbow method* used for calibrating the number of features can result in ambiguous choice of the cutoff because of the qualitative nature of the technique. To compensate for the inherent flaw of the elbow method, we could have also employed the *silhouette clustering*, a mathematical algorithm that provides the number of optimal clusters in a dataset.