Towards Machine Learning Interpretability for Tabular Data with Mixed Data Types

Prativa Pokhrel and Alina Lazar
Youngstown State University
Youngstown, OH, USA
ppokhrel03@student.ysu.edu, alazar@ysu.edu

Abstract

Gradient Boosting (GB) algorithms have been proposed for a variety of automated predictions and classification tasks with applications in many domains. These methods work faster and provide superior performance compared to deep learning methods when applied to tabular datasets. Another advantage is their interpretability. There are many machine learning methods that can train tabular data successfully, however the inner workings are usually hidden to the user. In this context, SHAP values combined with GB methods, increase model transparency and provide not only consistent feature rankings but also show the contributions of the predictors for individual instances. In this work we train multiple GB models using several tabular datasets and compare the result in terms of speed, performance and the global and local models' interpretability.

Introduction and Background

Advances in computer and storage technologies resulted in an exponential growth of the data collected and stored. Using available data and taking data-driven decisions gives businesses an advantage against the competition. The problem has shifted from collecting large amounts of data to mine and understand them, transforming them into knowledge, decisions, and actions. Lately, machine learning is becoming the dominant method of analyzing data. Machine learning models [3, 10] are now being used to solve a variety of real-world problems in different disciplines, ranging from retail and finance to medicine and healthcare, that requires high predictive accuracy.

However, this improved predictive accuracy has often been achieved through increased model complexity which leads to a lack of transparency. The "hows" of models' predictions are usually hidden to the user and this prevents the human expert being able to check the correctness behind the reasoning of the model. The quality and quantity of data used to train machine learning (ML) algorithms are directly related to their predicted ability [10]. Accurate predictions can lead to better explanations. In fields such as healthcare and engineering, it is crucial to understand how predictions are made.

The interpretability of machine learning models is critical for data scientists, researchers and developers, not only for understanding the inner working of models, but for debugging purposes and eliminating models' bias. As the use of machine learning algorithms in high-stakes domains have a significant effect on people's lives it is becoming essential to push the requirement for interpretability. For this study, we create a suite of interpretable GB models and we explore ways to understand the inner working of models whilst preserving the high predictive performance levels. We will focus on comparing GB models for different tabular datasets with mixed data types rather than creating a new models.

Methods

Two tabular datasets are trained using different machine learning algorithms and their performances are measured and compared using the logloss measure. The algorithms selected to train the datasets include Linear, Random Forest, LightGBM, Xgboost, CatBoost, Neural Networks, Nearest Neighbors, Decision Tree, Extra Trees, and Ensemble. The models based on the gradient boosting algorithm (LightGBM, Xgboost, Catboost) showed the best performance in terms of logloss and accuracy. It has been shown recently [11] that gradient boosting model outperform deep learning models in terms of training time and accuracy for both classification and regression. Therefore, in this paper we focus on exploring Gradient Boosting methods and their interpretability.

XGBoost [4] is a generalized GB method based on decision tree ensembles, that has proven to be an accurate and a fast machine learning approach for classifying tabular data. XGBoost training minimizes an objective function combining training loss and a regularization term. Regularization together with randomization techniques control the complexity of the model and reduce overfitting. The XGBoost algorithm features an efficient implementation for computing the best node split to speedup the slowest part of algorithm [2].

LightGBM [6] is a method that improves on XGBoost by providing faster training, better accuracy, less memory requirements by using a highly selective sampling procedure. Same as XGBoost, LighGBM employs the precomputed histogram of features. Unlike XGBoost that grows the trees level-wise, LighGBM builds the trees leaf-wise or vertically. In addition to the basic gradient boosting implementation, LightGBM features many types of randomizations,

including column permutations and bootstrap subsampling.

CatBoost [9] is the gradient boosting algorithm design to automatically deal with categorical features, by substituting each categorical feature with a numeric feature that measures the expected target value of each category. The gradients are updated using an upgraded procedure to avoid the prediction shift that occurs during training on different subsets. The implementation of CatBoost provides good results out of the box without any additional tuning. The algorithm builds a single model per iteration [2].

Mljar-supervised along with **SHAP** (SHapley Additive exPlanations) are used as a pipeline for model and hyperparameter selection, training, testing, feature importance and interpretability. The mljar-supervised is an Automated Machine Learning (AutoML) Python package for tabular data. The AutoML framework optimizes ML workflows and their hyperparameters in order to save time for the data scientist [12] and help non-experts. In addition to the optimization part, mljar, includes the SHAP explanations in the package. SHAP is a game theoretic approach used to explain the output of tree-based machine learning models. At the global dataset level it can be used to find the most important features through summary plots. At the local level, dependence plots, show how single features affect the predictions made by the model [5]. SHAP has been selected for this study above other similar tools such as LIME because SHAP values prove more consistent with human intuition [8].

Datasets

We used two tabular datasets that have features of different types. Tabular datasets is collections of rows and columns where columns represent the features while rows are the actual values. The datasets contain mixed type data which is a combination of both categorical and numerical data types. Numerical data type is the type of data that is expressed in terms of numbers rather than categorical descriptions for example, age, weight, height etc. On the other hand, categorical data type is a type of data that can be stored into groups or categories for example, ethnicity, gender, hair color etc. The datasets are as follows:

Adult Census Income is tabular dataset with 14 attributes: age, workclass, education, marital status etc. It has 48842 rows and a mixture of 8 categorical features and 6 numerical features. The binary classification task is defined as using demographic information to predict income bracket. This data, extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker [7], is pre-cleaned but still includes missing values.

In-vehicle coupon recommendation [13] is a dataset, with 25 features, 18 categorical and 7 numerical. It includes features such as destination, age, weather, gender, temperature, and marital status, and has 10147 rows total. The classification goal is to predict whether a person will accept an offered coupon or not. This data was collected via a survey made available on Amazon Mechanical Turk. The survey presents the interviewed person with a description of different driving scenarios and asks whether they will accept the offered coupon or not.

Experiments and Results

We train adult census income and invehicle coupon recommendation datasets with 10 algorithms using MLJAR and select the best model and parameters for each dataset. The performance of models are compared based on the log-loss evaluation metric and training time. Log loss is a global metric that represents the performance of a model. Further away the predicted probability is from the actual value, the higher its log-loss value is. From the log-loss boxplots, we find that Gradient Boosting algorithms perform better than other algorithms. As seen in the Figure 1 and Figure 2, Xgboost and LightGBM are the best models for Adult and Invehicle dataset respectively. The hyperparameters for the two best models are shown in Table 1.

To analyze the models' predictions we apply SHAP procedure to the best GB models. First, the feature importance plots show the features ranked from the most important to the least. Second, the SHAP dependence plots present a visual representation of how much change in feature's value changes the output of the model for that.

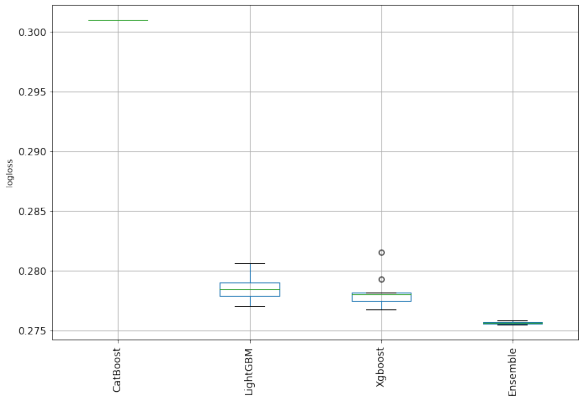

Figure 1: Adult Census Income logloss boxplot

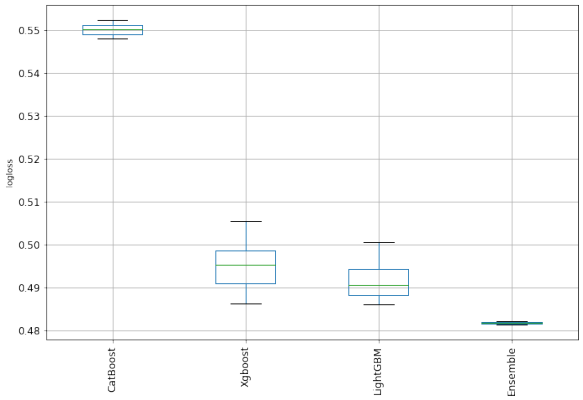

Figure 2: Invehicle Recommendation logloss boxplot

Table 1: Best models and parameters

Adult Income	Invehicle
Xgboost	LightGBM
objective: binary:logistic	objective: binary
eta: 0.075	num_leaves: 63
max_depth: 6	learning_rate: 0.05
min_child_weight: 1	feature_fraction: 0.9
subsample: 1.0	bagging_fraction: 0.9
colsample_bytree: 1.0	min_data_in_leaf: 10
eval_metric: logloss	metric: binary_logloss

Feature Importance

SHAP feature importance ranking is an alternative to permutation feature importance. The major difference between the two feature importance measures is that permutation feature importance is based on the decrease in model performance score, whereas SHAP importance is based on the magnitude of the shaply values. From the SHAP feature importance barplot in Figure 3 and 4 we can say that for the adult census income dataset, marital status and age are the most important features followed by capital gain and education. In the case of in-vehicle coupon recommendation, coupon type (coupon for coffee house or a restaurant or a bar), expiration and CoffeHouse have the highest impact out of all features on the prediction output.

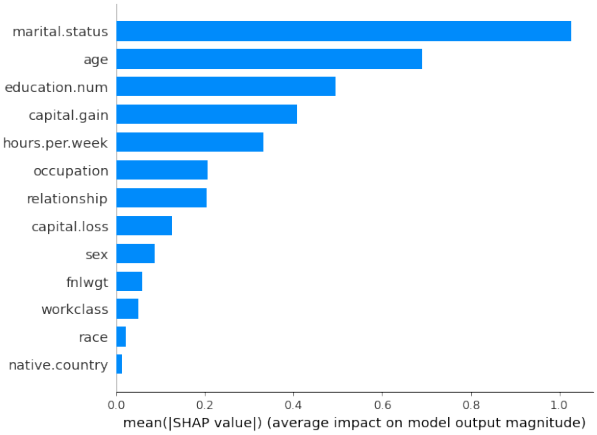

Figure 3: Adult census income feature importance

The feature importance plot is useful, but does not offer much more information about the model than the importance of the features. For more information, we analyze the dependence plots. The SHAP dependence plots show the effect of one feature or maximum two features on the model's predictions. On these plots, each dotted point represent one instance from the dataset. The horizontal axis (x-axis) is the value of the feature in consideration. The vertical axis (y-axis) stands for the SHAP value of that feature. These shapley values represent how much a feature's value changes the output of the model for that particular instance's prediction.

The two-way partial dependence plots for adult census

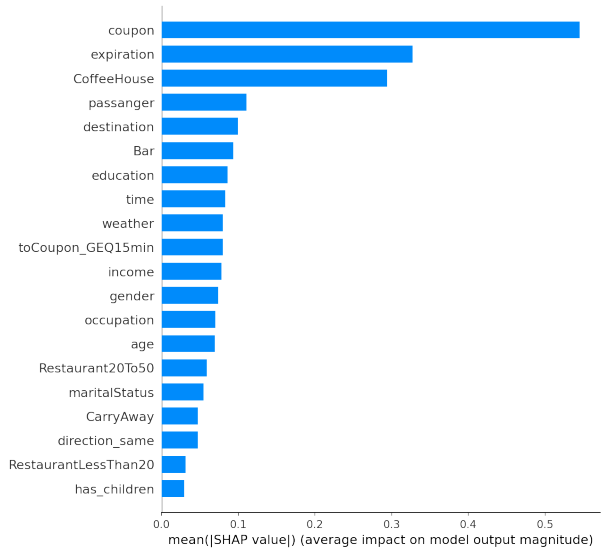

Figure 4: Invehicle coupon dataset feature importance

income suggest that middle-aged people with higher education (Figure 5) and people with higher education that are unmarried (Figure 6) have the higher chances of earn more money(>$50k). From the invehicle coupon recommendation two-way partial dependence plots we can imply that drivers who have not used a coupon in the last month have a high chance of accepting one (Figure 7) and a coffeehouse coupon which expires in 2 hours is more likely to get accepted (Figure 8) .

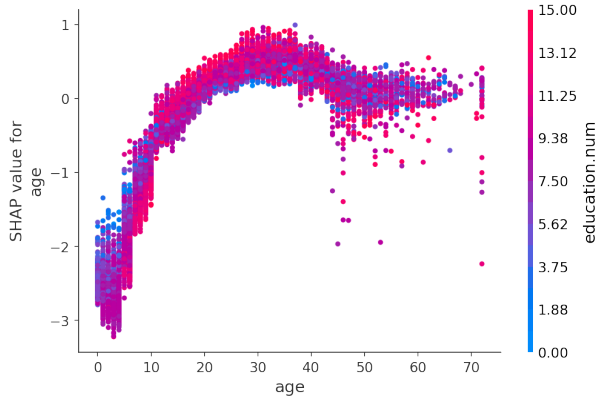

Figure 5: Dependence plot for age colored by education

Conclusions

From the logloss plots we observe that Gradient Boosting algorithms (Xgboost, LightGBM, CatBoost) perform better than other algorithms. Building an ensemble of GB models improves the accuracy even further. From the SHAP feature importance plots we conclude that the top ranked features in terms of importance for the adult census income dataset and in-vehicle coupon recommendation dataset. Next, we show

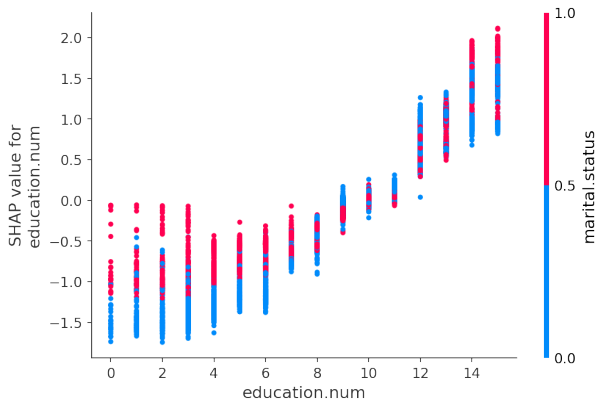

Figure 6: Dependence plot for education colored by marital status

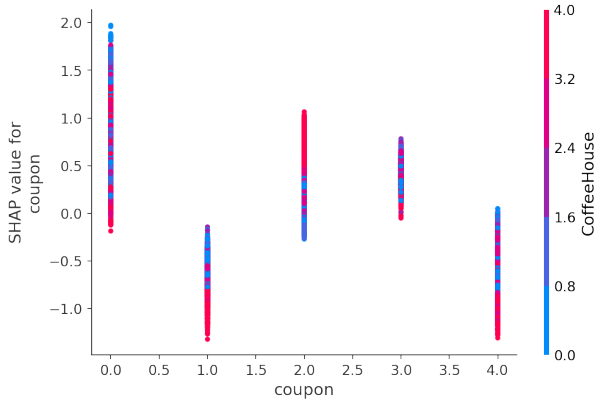

Figure 7: Dependence plot for coupon colored by coffeehouse

on the SHAP dependence plots how some of the high ranked features influence the prediction of individual instances.

In future we plan to investigate the performance of the Catboost Gradient Boosting algorithms and compare it to Xgboost and LightGBM. Also, we plan to train the datasets with the deep learning based method TabNet [1]. Later, we will compare TabNet performance with the GB methods' performances.

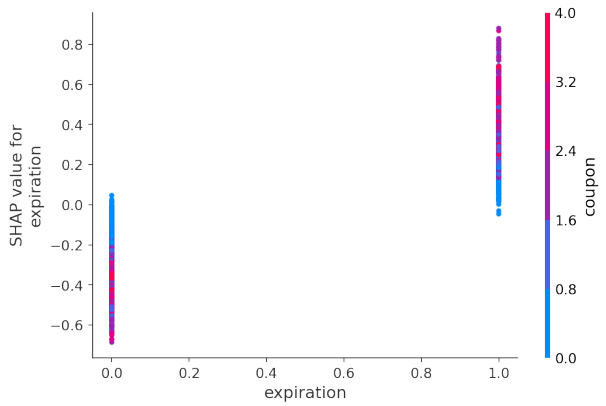

Figure 8: Dependence plot for expiration colored by coupon

References

[1] Arık, S. O., and Pfister, T. 2021. Tabnet: Attentive interpretable tabular learning. In *AAAI*, volume 35, 6679–6687.

[2] Bentéjac, C.; Csörgő, A.; and Martínez-Muñoz, G. 2021. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review* 54(3):1937–1967.

[3] Carvalho, D. V.; Pereira, E. M.; and Cardoso, J. S. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8(8):832.

[4] Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

[5] García, M. V., and Aznarte, J. L. 2020. Shapley additive explanations for no2 forecasting. *Ecological Informatics* 56:101039.

[6] Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30.

[7] Kohavi, R., et al. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, 202–207.

[8] Lundberg, S. M., and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30.

[9] Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31.

[10] Sahakyan, M.; Aung, Z.; and Rahwan, T. 2021. Explainable artificial intelligence for tabular data: A survey. *IEEE Access* 9:135392–135422.

[11] Shwartz-Ziv, R., and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion* 81:84–90.

[12] Truong, A.; Walters, A.; Goodsitt, J.; Hines, K.; Bruss, C. B.; and Farivar, R. 2019. Towards automated machine learning: Evaluation and comparison of automl approaches and tools. In *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI)*, 1471–1479. IEEE.

[13] Wang, T.; Rudin, C.; Doshi-Velez, F.; Liu, Y.; Klampfl, E.; and MacNeille, P. 2017. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research* 18(1):2357–2393.