Customer Classification using XGBoost: Accurate & Scalable Prediction of Customer Cluster Membership

Ewa Nowakowska* Joseph Retzer†

Abstract

High dimensional data analysis for predictive model development is both challenging and valuable. Various predictive models, e.g. CART, Random Forest analysis, bagging, neural networks, support vector machines, etc., have been shown to provide useful information, under various circumstances, for out-of-sample prediction. An alternative approach, known as stochastic gradient boosting (see [4]), has demonstrated remarkable results and is therefore often the preferred choice for predictive modeling.

All afore mentioned methods however can be rendered ineffective when working with very large or high dimensional data. In other words, these methods tend to not "scale" well and in addition "over fit" when applied to data sets with many variables. In this paper we will employ "XGBoost" (eXtreme Gradient Boosting), developed by Tianqi Chen and Carlos Guestrin of the University of Washington, for categorical response prediction. XGBoost provides a regularized, scalable and flexible, i.e. customizable and tunable, implementation of gradient boosting. This presentation will begin with a brief intuitive overview of ensemble-based boosting, culminating in its latest incarnation, extreme gradient boosting (XGBoost). This paper illustrates the ease of XGBoosts implementation in R through its application to the prediction of customer segment membership.

In order to insure XGBoost provides comparatively superior predictive performance, it is highly advisable to "tune" the model through appropriate parameter value selection. This paper therefore also outlines optimal parameter selection for the XGBoost model using the R "caret" package.

Key Words: XGBoost, segmentation, database scoring, Friedman

1. Overview

1.1 Business / Analytics Goals

The focus of this research is to develop a marketing strategy which addresses the needs and concerns of investor consulting services participants. Specifically, we intend to uncover appropriate customer (participant) segments for targeted messaging in order to create long-term consulting participant strategies.

The analytics involved in accomplishing the above are outlined as follows:

- Step 1: Identify and profile latent participant segments. Segments should be both of high quality and predictable using corporate database information.
- Step 2: Develop an effective predictive algorithm on cross-validated cases from data used to identify the segments.
- Step 3: Apply the algorithm selected in Step 2 to score the larger participant base.

*Associate Partner, Data Science, EY
†ACT Market Research Solutions

1.2 Cluster Analysis

The goal of the unsupervised learning (cluster) analysis is to identify segments which are of "high quality", i.e.,

- homogeneous within and
- heterogeneous between

Cluster quality may be evaluated with various metrics including silhouette plots. Silhouette plots were employed in this study to provide a graphical, segment specific, profile of quality.

The algorithm should also produce clusters which profile well on internal, corporate database metrics. To accomplish both goals simultaneously, a variety of clustering algorithms were applied and evaluated. An approach known as "Semi-Supervised Learning Cluster Analysis" (see [6]) produced results superior to all others investigated.

1.3 Database Scoring

The next step involves choosing an effective predictive model by accessing its out of sample predictive accuracy. As with the cluster analysis, a variety of models were evaluate including:

- Linear Discriminant Analysis
- Multinomial Logit
- Random Forest ([1])

The Random Forest approach produced a more than acceptable level of accuracy (76%) based on a sample of size 7,226 participants. As cluster quality was also high, the sense was that this performance would be difficult to surpass. However, since the size of the database to be scored was roughly 1.5 million investors, it was clear that even a "small" improvement in accuracy would entail a significant increase in the absolute number of correctly scored investors. For that reason additional machine learning algorithms were employed and evaluated. As the highest predictive accuracy was attained using "eXtreme Gradient Boosting (aka XGBoost)", this approach was used in place of Random Forest Analysis to score participants.

2. Scoring Algorithm

2.1 XGBoost (see [2], [3])

XGBoost was first investigated based on its reputation of being a both scalable & accurate approach to predictive modeling. In addition, it is the 2016 John M. Chambers Statistical Software Award winner. Also, its superior performance in the online competitive predictive analysis platform, Kaggle, is evidenced in the following quote,

> *"As long as Kaggle has been around, it has almost always been ensembles of decision trees that have won competitions.*
> *It used to be* **random forest** *that was the big winner, but over the last six months a new algorithm called* **XGBoost** *has cropped up, and*

its winning practically every competition in the structured data category.

For non-structured data, the winning machine learning methods seem to be forms of artificial neural networks."

– Anthony Goldbloom,
founder and CEO of Kaggle.

2.2 Boosting

An overview of XGBoost requires first examining the more general "boosting" approach to ensemble modeling. To this end, a conceptual understanding of "boosting" aids in reconciling the numerous approaches to boosting, e.g.,

- **Ada-Boost** (Adaptive Boosting)
- **Gradient Boosting**
- **Gradient Tree Boosting**
- **lightGBM, CatBoost, LPBoost, TotalBoost, BrownBoost, MadaBoost** ...

While each model differs, sometimes significantly, in its approach, there is in fact an underlying idea central to all. Specifically, each involves the modeling of errors found in the previous ensemble member (decision tree). Once boosting is viewed in this manner, the relationship among the algorithms above becomes clear.

The XGBoost implementation of boosting may be described as a heavily regularized, single machine multi-level parallelized implementation of gradient tree boosting. Unlike the "gradient boosting machine" approach however, XGBoost's regularized model formalization controls over-fitting, providing significantly superior performance.

The objective function (for purposes of illustration and SSE loss function with regularization term) may be written as:

$$\text{obj}(\theta) = l(\theta) + \Omega(\theta) \qquad \text{where} \qquad l(\theta) = \sum_{i}^{n} (y_i - \hat{y}_i)^2 \qquad \text{and}$$

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F} \text{ where } \begin{cases} K = \text{number of trees} \\ \mathcal{F} = \text{space of all possible CART's} \end{cases}$$

The unique component of XGBoost is the regularization term, $\Omega(\theta)$, expressed below:

$$\Omega(\theta) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Where:

- w is the vector of scores on leaves and T is the number of leaves.

- Term is a combination of L2 norm and number of leaves, T.
- Various ways to specify complexity but above works well in practice.

The "Extreme" in XGBoost refers to the engineering of pushing computational resource limits for boosted tree algorithms. XGBoost accomplishes this through:

- Parallelized split-finding,
- the computational component of XGBoost is written in C++.
- XGBoost pre-processes data before training algorithm and
- Memory & cache-line optimization is implemented.

2.3 XGBoost Tuning

A critically important component in the development of an XGBoost model is "model tuning". Tuning involves a "grid search" across multiple parameter vector specifications culminating in the optimal selection based on k-fold cross validation model accuracy results. Tuning is accomplished using the R package "`caret`" (Classification And Regression Training (CARET) [5]) as detailed in Figure 1 below:

1. Define sets of model parameter values to evaluate
2. **for** *each parameter set* **do**
3. **for** *each re-sampling iteration* **do**
4. Hold-out specific samples
5. [Optional] Pre-process the data
6. Fit the model on the remainder
7. Predict the hold-out samples
8. **end**
9. Calculate the average performance across hold-out predictions
10. **end**
11. Determine the optimal parameter set
11. Fit the final model to all training data using the optimal parameter set

Figure 1: R-Caret Tuning Algorithm.

3. Results & Summary

3.1 XGBoost Accuracy

Predictive performance of the XGBoost scoring model attained an accuracy of 80% correct and is graphically illustrated in the confusion matrix visualization, Figure 2, below:

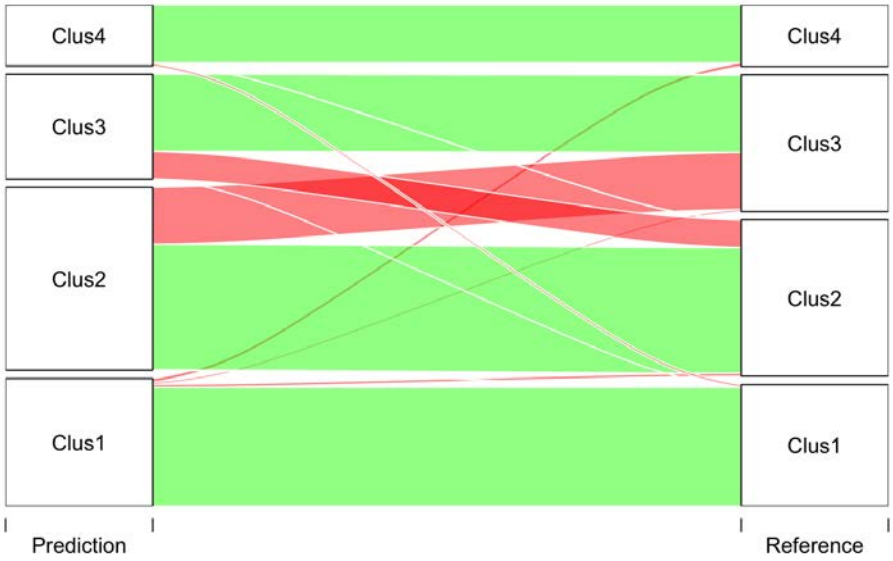

Figure 2: XGBoost Scoring Results Confusion Matrix.

The confusion matrix suggests mixing in prediction vs. actual primarily between the 2^{nd} and 3^{rd} clusters while clusters 1 and 4 exhibit little predictive error.

Attribute importances may also be produced from the XGBoost model (see Figure 3 below) for additional insight into model results.

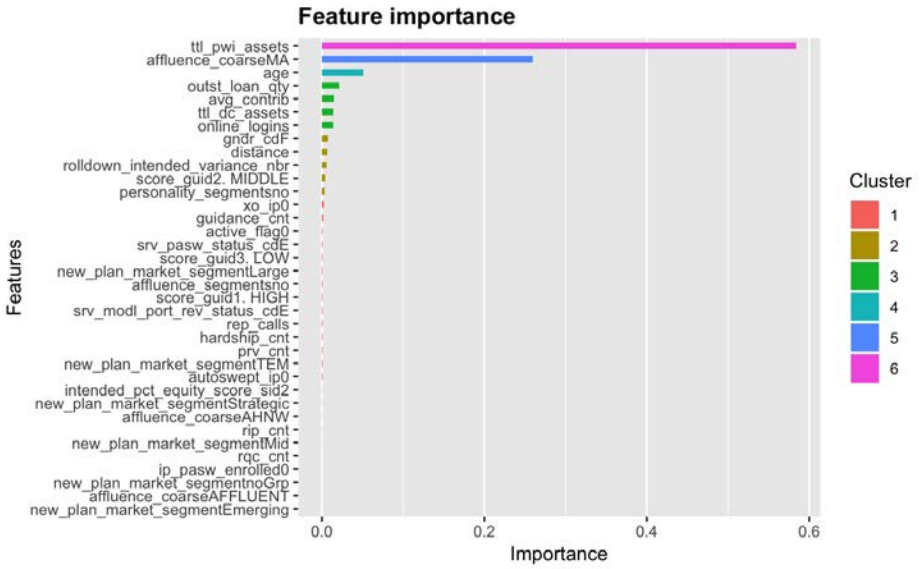

Figure 3: Feature Importance: (`xgb.ggplot.importance`).

Importance is measured as the improvement/gain in the performance measure

(objective function), weighted by the number of observations in the node. In addition, importances are clustered using the R package `Ckmeans.1d.dp`.

3.2 Summary

The analytics goal of this research was to identify high quality clusters which are predictable using corporate internal metric data. This was accomplished by applying the "SSL Cluster Analysis" algorithm as described in [6].

Cluster membership was initially predicted using a variety of approaches with Random Forest predictive modeling proving to be best. Since the size of the customer base was relatively large however (approximately 1.5 million), finding even a marginally more accurate algorithm has definite value. This is due to the fact that even a small change in predictive accuracy translates into a potentially large change in the absolute number of customers classified appropriately.

To that end, XGBoost was investigated and is outlined in this paper. The XGBoost algorithm provided and accuracy rate of approximately 80%. The 4% increase in accuracy (over the 76% produced by the Random Forest model) translates into $.04 \times 1.5 million = 60,000$ customers making the application of XGBoost a worthwhile alternative.

References

[1] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, oct 2001.

[2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[3] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Li. *xgboost: Extreme Gradient Boosting*, 2019. R package version 0.81.0.1.

[4] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[5] Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, and Tyler Hunt. *caret: Classification and Regression Training*, 2019. R package version 6.0-84.

[6] Ewa Nowakowska and Joseph Retzer. Extending cluster ensemble analysis via semi-supervised learning. In *Proceedings of the Sawtooth Software Conference*, pages 251–266, Provo, UT, USA, 2013. Sawtooth Software.