Testing Differences in Performance of Pricing Models

Jose Pliego San Martin

February 6, 2023

Abstract

Pricing models play an important role in the insurance industry, and companies aim to price policies correctly by predicting the costs associated to them. Gradient Boosting Machines have shown great predictive performance but, unfortunately, are very unstable. Instability is an undesirable property in pricing models because it is often difficult to tell when one model is better than another or if the observed differences are a consequence of noise. The goal of this project was to implement a paired t-test to tell whether observed differences are statistically significant or not. The test is performed using multiple observations obtained through repeated cross-validation.

1 Introduction

Accurately pricing policies is key for insurance companies. Under-pricing policies results in losses for the company, while over-pricing policies may result in customers turning to competitors for a better quote. It is also important for pricing models to be stable. We can think about stability in three main aspects:

1. **Performance.** Is the difference in performance between models real or a consequence of randomness in the modeling pipeline?

2. **Predictions.** Are predictions consistent across different periods and model refits?

3. **Pricing Logic.** Are the features affecting the price as expected?

Focusing on performance stability, it is important for data scientists to tell when one model is better than another. Data scientists choose models based on their predictive performance. However, due to regulations or business considerations, they are sometimes forced to choose a worse performing model. It is then important to consider whether the performance differences observed are statistically significant.

The goal of this project was to implement a statistical test to make a decision on whether these differences are significant or not. In section 2 the idea of repeated

cross-validation is introduced along with the paired t-test introduced by Nadeau & Bengio (2003) to test differences in observations obtained through cross-validation. Section 3 shows four examples of how the test performs in actual pricing models, and section 4 talks briefly about the implementation and possible extensions.

2 Methods

There are four main sources of randomness in GBMs over which data scientists have some control: the way data is split into training and validation sets, the randomness in some hyperparameter tuning procedures, the bootstrap samples that GBMs take at each step, and the feature sampling also done at each step. After multiple experiments, it was found that most of the variability in performance comes from splitting data and tuning hyperparameters. The other two sources of randomness (bootstrap and feature sampling) are almost negligible for performance purposes. It is interesting to note that the last two sources identified are also the only ones specific to GBMs. Therefore, the methods introduced here can be useful for any Machine Learning algorithm.

A common way to get more robust estimates while accounting for randomness in the data splitting procedure is cross-validation (Hastie et al., 2017). Cross-validation consists of splitting the data into k different disjoint sets and using each of the k sets as a validation set once and the other k-1 as a training set. In the end, we have k different models and therefore k different performance observations.

The main disadvantage with cross-validation is that the models are not independent. This is because every observation serves as either a training or validation record in all the models. It is then important to choose a "good" number of folds. Choosing too many folds results in lower bias because the models are trained on more data, but variance increases because models are more correlated. On the other hand, having not enough folds means that variance decreases because the models are less correlated, but bias increases since models are trained on a smaller data set.

To avoid some of the issues when choosing the number of folds, the idea of repeated cross-validation is to perform k-fold cross-validation (described earlier) a total of r times (Bouckaert & Frank, 2004). In the end, a total of $r \times k$ performance observations are obtained and can be used to get a more robust estimate of performance and measure variability around it.

Finally, a paired t-test can be performed with the $r \times k$ performance observations of two different models. It is important to consider that the observations are not independent and therefore classical t-tests yield high type I error probabilities. The test introduced by Nadeau & Bengio (2003) aims to stabilize type I error probabilities

by adjusting the variance estimate. The statistic used is

$$t = \frac{\sum_{i=1}^{r} \sum_{j=1}^{k} x_{ij}}{\sqrt{\left(\frac{1}{k \times r} + \frac{n_2}{n_1}\right) \hat{\sigma}^2}}, \tag{1}$$

where r is the number of repeats, k is the number of folds in each repeat, n_2 is the number of observations in each of the k validation sets, n_1 is the number of observations in each of the k training sets, and x_{ij} is the difference in performance between models fit in fold j of repeat i.

Note that, as mentioned earlier, the statistic in equation 1 has an extra term in the denominator to adjust for correlation in the observations. This statistic has been shown through simulations to provide stable type I and type II error probabilities (Bouckaert & Frank, 2004).

3 Results

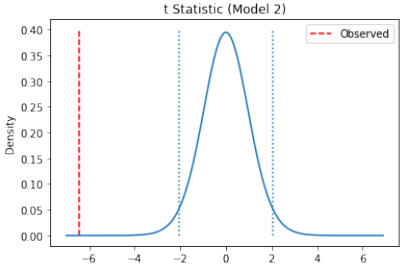
The procedure from section 2 was tested with four different models. The observed statistics for each model are shown in figure 1 along with the t-distribution. All tests were performed with 5×5 repeated cross-validation and 0.05 type I error probability.

Panel 1a shows that the statistic falls well within the rejection region after removing the most important predictor in the pricing model, suggesting that there is evidence to conclude that the model with all the predictors performs significantly better than the model without the predictor.

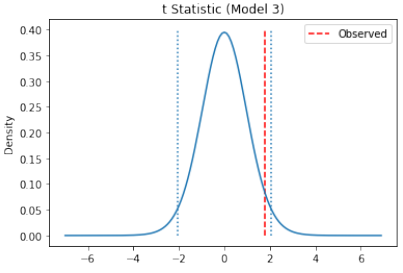
Panel 1b shows that the statistic falls outside of the rejection region after adding back a predictor that the team previously considered not important. Note that the statistic falls close to the region border. Choosing beforehand a different type I error probability may have resulted in the test rejecting the hypothesis that both models perform equally well.

Finally, panels 1c and 1d show that the test suggests that there is no significant change in performance after removing the least important predictors according to two different importance metrics (gain importance and permutation importance).

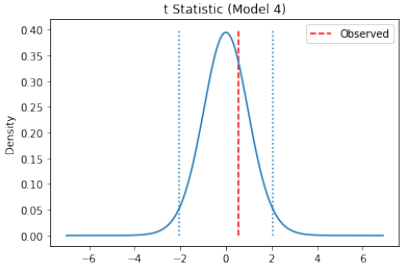
The examples shown in this section were included as sanity checks to make sure that the procedure yields the expected results in some cases that can be described as "easy".



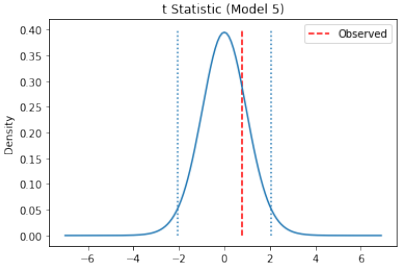
(a) Removing the most important predictor.



(b) Adding a non-important predictor.



(c) Removing a non-important predictor.



(d) Removing a non-important predictor.

Figure 1: Observed statistics (1) in four different cases.

4 Conclusion

All the functions needed to perform the hypothesis test were incorporated into the internal Python package that the team uses to fit pricing models. Functions were coded to create the repeated cross-validation data sets, upload them to the database management system, create the configuration for the $r \times k$ models and train them in parallel. The idea was to offer the team the necessary tools to easily incorporate this procedure into the regular modeling pipeline.

Some of the extensions discussed with the team consist of taking advantage of the several models available after performing repeated cross-validation. Robust estimates and variability measures can be obtained in other aspects like variable importance rankings for feature selection, Shapley values for model interpretability, and hyper-parameter tuning.

References

Bouckaert, R. R., & Frank, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. In H. Dai, R. Srikant, & C. Zhang (Eds.) *Advances in Knowledge Discovery and Data Mining*, (pp. 3–12). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2 ed.

Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*(3), 239–281.
URL `https://doi.org/10.1023/A:1024068626366`