ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter56872020.pdf
https://doi.org/10.30534/ijeter/2020/56872020

The Sensitivity Conundrum – Random Forest or Boosting

Dr.Dhimant Ganatra[1], Dr.Dinesh Nilkant[2]
[1]CMS Business School, JAIN (Deemed-to-be University), Bangalore-560009, India, dr.dhimantganatra@cms.ac.in
[2]CMS Business School, JAIN (Deemed-to-be University), Bangalore-560009, India, dineshnilkant@cms.ac.in

ABSTRACT

In the classification context, tree-based models are simple and useful for interpretation. However when it comes to model accuracy the single-tree model does not match the power of other supervised learning approaches. By aggregating trees a model's accuracy can be improved. Ensemble methods like random forest and boosting combine predictions from multiple models into one that is far superior to the individual models. Depending on the business goal, the accuracy paradox may come into play. The class statistics, precision and recall may be more important than the overall accuracy. The True Positive Rate varies based on the type of ensemble used among other factors. While both random forest and boosting lead to some loss of interpretability, the improvement in sensitivity will outweigh this loss. By tuning several of the model's parameters it is possible to achieve higher levels on any of the four counts in the confusion matrix. While both random forest and boosting use trees as base learners, they differ primarily in the way the trees are built. A comparison of both the approaches is made to identify a superior performer on the positive class.

Key words: Random Forest, Boosting, Machine Learning, Sensitivity.

1. INTRODUCTION

1.1 The ClassificationTask

There is reason to believe that B-Schools will see an increase in applications in the ongoing 2020 and the following 2021 season. The general slowdown in the economic activities will have an impact on the placement in case of students completing their under-graduation in these two years leading to an increased demand for MBA. MBA is seen as an investment during recession and slowdown [1]. According to Graduate Management Admission Council (GMAC), MBA Applications have increased in recessions.In India too this trend was noticed during the last economic slowdown in 2008 [2]. With an increased application base and limited seat availability, a B-School would want to be choosy in offering admission. A wrong pick will definitely impact the placement season two years down the line. Since the current economic slowdown is here to stay [3], the B-School under study was interested in knowing whether it is possible to differentiate the pool of applicants based on placeability. A B-School which has a lower acceptance rate into its program looks at many competencies when offering admission. Academic ability among others stands at the top.

Using the classification tree algorithm, a business rule was developed [4], which was able to classify applicants into two classes – Placeable and Not Placeable with 83.33% accuracy. The tree found acceptance with the admission team as it was easily interpretable and closely mirrored the human decision-making process in general. Unfortunately, the admission team was not happy with the model's overall accuracy and more specifically sensitivity which is 64.71% in the base model. The positivity rate which is in news almost every day during the Covid-19 pandemic has found fascination with the B-School also.Bagging and random forest ensembles were applied to improve the model's sensitivity [5]. The objective of this research paper is to build further on the model as proposed in [5], to improve its accuracy by avoiding misclassification of positives which in this case means wrongly identifying a non-placeable applicant as placeable. Using ensemble methods like random forests, and boosting, the accuracy of a single-tree classifier can be improved [6]. In the process, a comparison is made between the Boosting and random forest algorithm.

1.2 Ensemble Method

Ensemble learning is a powerful technique to improve the performance of a machine learning model.When we use a traditional machine learning algorithm to train a model in most of the cases we find that we don't get good accuracy. In predicting a target variable, the main causes of difference in actual and predicted values are noise, variance, and bias.Ensemble learning techniques attempt to make the performance of the predictive models better by decreasing variance, bias or improve predictions. It is an art of combining several individual learners to improvise on the prediction power.The fundamental principle of the ensemble model is that a group of weak learners come together to form a strong learner, which in turn increases the accuracy of the model [7]. Although there are several types of ensemble learning methods, they can broadly be divided into two groups – parallel ensemble methods like random forest where the base learners are generated in parallel and sequential ensemble methods like boosting which involve generating learners sequentially. Which one to use depends on the problem at hand.Most ensemble methods use a single base learning algorithm to produce homogeneous base learners, i.e. learners of the same type, leading to homogeneous ensembles.Ensemble methods can be used in both classification and regression setup. When used in classification, the multiple classifiers that are developed are likely to classify a new observation in different categories. Then a strategy of majority voting is used to decide the final class of the new observation. Such majority voting could be based on simply counting the vote from each class or could be weighted based on accuracy. In case of regression problems, the prediction of a new observation is simple average or weighted average of all the predictions from the set of the developed regression models [8]. In short, an

ensemble learning modelis a general-purpose procedure for reducing the variance of any statistical learning method [9].

2. LITERATURE REVIEW

Decision tree algorithm has been used for decision-making in an educational setup for long and there have been many studies on application of the classifier algorithm in identifying potential academic parameters. These studies have focussed on measuring student's academic outcome, but none have connected admission to placement for a B-school. Same is the case for ensemble methods. Classifier models have been developed to understand student success in exam and course completion using academic features. There are numerous research papers on predicting student placement but most of them use complex algorithms and are built on data post enrolment into a course. Reference [10] shows how the use of ensemble methods provided better results in an e-learning setup. An ensemble meta-based tree model classifier technique for predicting the student performance was used by [11]. The proposed model essentially combined two consistent machine learning techniques into a voting bagging technique to achieve higherperformance. Ensemble techniques based on four representative learning algorithms were used by [12] to construct and combine different number of ensembles to predict whether a student will be able to successfully complete his degree. The performance of Adaboost and Bagging ensembles were better than Random Forest. In [13], Bagging and Boosting ensembles were evaluated on 23 datasets with decision tree as the classifier algorithm. Findings suggest that most of the gain in an ensemble's performance comes in the first few classifiers combined. Reference [14] concluded that building a random forest of trees improves the classifier. 10 years of data of a University were used to predict whether a student will complete the degree based on their performance in courses of first two semesters. Classification models based on the gradient boosting was created by [15] to predict academic outcomes of student performance at the end of the school year using decision tree as the base classifier. Ensemble model developed in [16] provided increased accuracy in identifying students who are likely to fail or may drop out. In [17], comparisons of AdaBoost and random forest algorithms on their ability to perfectly fit the training data have been done in a wide variety of situations. It was conjectured thatboosting should be used like random forests, with large decision trees, without regularization or early stopping.

3. PARALLEL AND SEQUENTIAL METHODS

These methods differ in terms of how data is selected from the weak dataset, how the weak models are generated, and how the outputs are combined to form a stronger classification tree model.

3.1 Random Forest

Random Forest is a popular ensemble method in which several trees are developed using simple strategies like Bagging. Each tree in the ensemble is built from a sample drawn with replacement,that is, a bootstrap sample, from the training set. In addition, instead of using all the features, a random subset of features is selected, further randomizing the tree.By forcing each split to consider only a subset of the predictors,random forests overcome the problem of tree correlation. This small tweak provides an improvement over bagging. Each time a spilt is considered, a random sample of only m out of p predictors is considered. The split is then based on only one of the m predictors thereby decorrelating the trees. A fresh list of m predictors is taken at each split and the value of m is approximately the square root of p. Where m equals p, then this amounts to bagging.Due to the limited number of predictors selected in each iteration, the generation of models is faster than bagging. In general, random forest approach is expected to provide much higher accuracy compared to a single tree [18].

3.2 Boosting

In bagging each tree is built on a bootstrap dataset independent of other trees. In boosting, the trees are grown sequentially meaning that each tree is grown by using information from previously grown trees. A training model concentrates on the misclassified records from previous models.That is, each tree in boosting is fit on a modified version of the original data setand then combining the classifiers via a weighted majority vote.Boosting frequently yields better models than bagging [9]. Boosting works on reducing bias thanvariance. Hence, it tends to improve upon its base models most when they have high bias and low variance.

Three most widely used algorithms in the boosting family are Adaptive Boosting (AdaBoost), Gradient Boosting and eXtreme Grading Boosting (XGBoost). While AdaBoost focusses on the misclassified records in subsequent classifiers, Gradient Boosting focusses on residuals from previous classifiers and fits a model to the residuals [19].Gradient boosting learns from the mistake - residual error directly, rather than update the weights of data points. XGBoost is one of the most widely used boosting algorithms. It optimises the construction of weak-learners thereby increasing speed and performance of gradient boosting algorithms.

4. METHODOLOGY

The objective here is to improve the accuracy especially of the positive class of the classifier model developed by [4]. The classifier model was built following the need of the B-School to differentiate among prospective students into placeable and non-placeable categories. Since the cost of misclassifying a non-placeable category is high, bagging and random forest ensemble methods were brought in by [5] to improve the existing model's sensitivity as well as overall accuracy. Given the current scenario of increased number of applicants for the MBA programme, it is felt that three is a further need to see the possibility of increasing a model's sensitivity even though it may come at the cost of interpretability. R language and environment (version 4.0.0) has been used for statistical computing and graphics [20].

215 students who completed their MBA from a Bangalore based B-School have been selected for the study as per [4].The dependent variable is a two-class categorical variable – Placement with labels as Placed and Not Placed. There are 10 predictor variables. The 75–25 technique is used [21] to split the data set. The crosstab in respect of the response variable is as shown in Table 1.

Table 1: Count of Train and Test set

	Placed	**Not Placed**	**Total**
Train set	111	50	161
Test set	37	17	54
Total	148	67	215

5. CLASSIFIER MODEL

5.1 Decision Tree Classifier – Base classifier

Recursive Partitioning And Regression Trees (rpart) library [22] provides the algorithm to create the decision tree.

Figure1 shows the classification tree for the train dataset. Nodes that split to the left are the ones which meet the criteria while nodes to the right do not. Each node is labelled by the predicted class, either Placed or as Not Placed. The percentage value is to be read from left to right, with the probability of Not Placed being on the left.

From Figure 1, at node 7 of the tree we understand that if a student has scored more than 56% in SSC and more than 66% in Degree, then there is more than 94% chance that the student is likely to be Placed and the support is 49%.

Table 2 lists out the results of the classifier model. The accuracy of classifying Placed (negative/specificity) is 91.89%, whereas the accuracy of classifying Not Placed (positive/sensitivity) is 64.71%. The overall accuracy is 83.33%. The positive and negative predicted values are 0.7857 (precision) and 0.8500 respectively. The classification accuracy of the test sample is within 10% of the training sample, this provides evidence of the utility of the model [23].Given the context, here, we need a higher accuracy in predicting positive classes rather than negative classes. The F-Measure which combines both precision and recall is 0.7097. Values closer to 1.0 are considered the best. ROC curve is used in order to understand the overall worth of a classification tree. The area under the ROC curve (AUC) is 0.8959, indicating the proportion of concordance pairs in the data. Models with higher AUC are preferred.

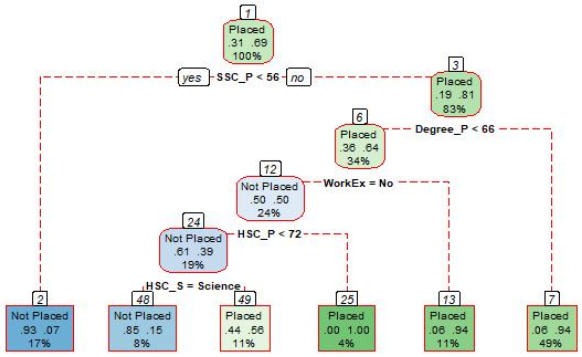

Figure 1: Classification tree for train dataset

5.2 Random Forest Ensemble

We first build a random forest ensemble with default hyperparameters – 500 trees and 3 predictors ($\approx \sqrt{p}$). This model leads to a reduction in both test error and OOB error over bagging. Table 2 lists outthe model parameters. The AUC is 0.9745 and can be considered good since it is very close the maximum of one.

It is possible to seek improvement to the model by tuning the hyperparameters. The most commonly tuned hyperparameters include – the number of trees to grow (ntree), number of variables randomly sampled as candidates at each split (mtry), and the size of the sample to draw for training. Table 2 shows the output for three such tuned models.It is possible to perform a larger grid search across many values of the hyperparameters by creating a grid and a loop through each possible combination. It is important that we choose an optimal set of hyperparameters to tune the model so as to better fit the data.

We have evaluated 3276 different models by varying the above three hyperparameters. The top 10 performing models have an OOB error between 11.80% and 13.04%which is lower than the default or the three manually tuned models. The best random forest model we found based on the grid search uses 180 trees, 4 variables, and a sample size of 63.2%. Table 3 shows the confusion matrix for the model applied to the test dataset.The AUC is an impressive 0.9571.

5.3 Boosting Ensemble

Boosting algorithms play a critical role in dealing with bias variance trade-off. From among the many boosting algorithms, we have focussed on AdaBoost, Gradient Boosting and XGBoost. AdaBoost creates its first decision stump, by weighing all observations equally. To correct the error, the observations that were incorrectly classified earlierwill now carry more weight than the observations that were correctly classified. The effect of varying the weight to give

Table 3: Confusion Matrix based on Random Forest

Predicted	Actual		Overall %
	Not Placed	Placed	
Not Placed	14	1	
Placed	3	36	
% Correct	82.35	97.30	92.59

Table 2: Random Forest Model Parameters

Technique	Hyperparameter		Train set			Test set	
	ntree	mtry	OOB Error	Sensitivity	Accuracy	Sensitivity	Accuracy
Classification Tree	--	--	--	0.7200	0.8882	0.6471	0.8333
Random Forest (Default)	500	3	16.15%	0.6200	0.8385	0.8235	0.9259
Random Forest (Tuned)	500	4	16.77%	0.6200	0.8323	0.8235	0.9259
Random Forest (Tuned)	500	6	16.77%	0.6200	0.8323	0.7647	0.9074
Random Forest (Tuned)	200	7	14.91%	0.6800	0.8509	0.7647	0.9074
Random Forest (Optimal m)	500	9	16.15%	0.6400	0.8385	0.7647	0.8889
Random Forest (Ranger)	500	3	17.39%	0.6000	0.8261	0.8235	0.9259
Random Forest (Ranger-best)	180	4	11.80%	0.7000	0.8820	0.8235	0.9259

Table 4: Boosting Model Parameters

Technique	Hyperparameter				Train set		Test set	
	n.tree	int.depth	n.minobs	shrinkage	Sensitivity	Accuracy	Sensitivity	Accuracy
AdaBoost	100	1	3	--	0.8200	0.9068	0.8824	0.9074
Gradient Boosting (Default)	100	1	10	0.1	0.6000	0.8509	0.7647	0.9074
Gradient Boosting (Tuned)	10000	3	5	0.001	0.7000	0.8882	0.8235	0.9259
Gradient Boosting (Best)	100	3	3	0.05	0.9000	0.9503	0.8824	0.9259
XGBoost (CV Tuned)	70	9	1	0.1	0.9000	0.9565	0.8824	0.9074

more emphasis to a difficult observation means that each subsequent machine has a disproportionately harder set of examples to train on. Since gradient boosting offers an advantage over the AdaBoost methodology, we have not delved in detail into the AdaBoost algorithm. The model output obtained using the adabag package [24] is as shown in Table 4.

In order to build a gradient boosting model, we first build the model using the default hyperparameters in the gbm package [25] – with 100 trees (n.tree), 1 as maximum depth of tree (int.depth), learning rate (shrinkage) of 0.1, and the minimum number of observations in the terminal nodes of the trees (n.minobs) being 10. Table 4 lists the model parameters. An improvement over this default model can be achieved by tuning the hyperparameters. The AUC for one such tuned model is 0.9507.

An easier way to perform a larger grid search on all the critical hyperparameters is by using the caret package. We have evaluated 2430 combinations of the four hyperparameters. The parameters obtained using the recommended hyperparameters are as shown in Table 4. Though the overall accuracy on the test set remains same, we see a considerable increase in the model's sensitivity which is our objective.Table 5 shows the confusion matrix for the model applied to the test dataset.

XGBoost[26] is similar to gradient boosting algorithm but it has a few tricks up its sleeve which makes it stand out from the rest. This algorithm uses multiple hyperparameters and as such tuning is a must. The optimal value for each of them has been arrived at by performing a simple cross-validation. XGBoost allows us to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. The results are as shown in Table 4.

Table 5: Confusion Matrix based on Gradient Boosting

Predicted	Actual		Overall %
	Not Placed	Placed	
Not Placed	15	2	
Placed	2	35	
% Correct	88.24	94.60	92.59

6. CONCLUSION

Boosting is a viable approach to reducingprediction error. It gives statistically significantimprovement in most cases and never isstatistically worse than random forest or bagging in general [27].

The best tuned boosting algorithm has increased the accuracy by 11% and sensitivity by a whopping 36% as compared to a single tree. The boosting algorithm has also fared better than its random forest counterpart. Though there has been no change in the overall accuracy, the sensitivity has increased by more than 7%.

We can obtain an overall summary of the importance of each predictor. Table 6 indicates the importance matrix relevant to the top 3 features for the model built using XGBoost. The Gain is the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction

Table 6: Feature Importance

Feature	Gain	Feature	Gain
SSC_P	0.4493	HSC_SScience	0.0257
HSC_P	0.2263	SSC_BOthers	0.0192
Degree_P	0.1043	Degree_TS&T	0.0106
WorkExYes	0.0675	HSC_BOthers	0.0104
Etest_P	0.0433	HSC_SComm	0.0055
GenderM	0.0379		

Future work: Future studies can examine the use of neuralnetworks or oblique decision tree as learningmachines, and the feasibility of using non-linearfunctions of features as inputs to decision trees.

REFERENCES

1. Economic Times (2020, February 24). **It pays to do an MBA in a Slowdown**. https://economictimes.indiatimes.com/jobs/placements-for-the-class-of-2020-non-bluechip-b-schools-beat-slowdown-blues/articleshow/74234601.cms
2. Economic Times (2009, January 21). **MBA is the best investment during recession**. https://economictimes.indiatimes.com/mba-is-best-investment-during-recession-execs/articleshow/4010028.cms?from=mdr
3. BBC News (2020, June 12). **Coronavirus: What is a recession?** https://www.bbc.com/news/business-52986863
4. D. Ganatra, and D. Nilkant.**A Business Rule for a B-School using Machine Learning**. *International Journal of Advanced Trends in Computer Science and Engineering*, vol 8, no. 6, pp. 3621-3627, Dec 2019.

https://doi.org/10.30534/ijatcse/2019/145862019
5. D. Ganatra, and D. Nilkant.**Ensemble Methods to improve Accuracy of a Classifier**. *International Journal of Advanced Trends in Computer Science and Engineering*, vol 9, no. 3, pp. 3434-3439, Jun 2020. https://doi.org/10.30534/ijatcse/2020/145932020
6. L. Breiman. **Random Forests**. *Machine Learning*, vol. 45, no. 1, Springer, pp. 5–32, 2001. https://doi.org/10.1023/A:1010933404324
7. G. Seni, andJ. F.Elder.**Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions**. Morgan & Claypool Publishers, 2010.
8. M. Pradhan, U. D. Kumar, **Machine Learning using Python**. New Delhi: Wiley India, 2019, pp. 225-239.
9. G. James, D. Witten, T. Hastie and R. Tibshirani. **An Introduction to Statistical Learning with Applications in R**. Springer, 2017, ch.8.
10. P. Kumari, P. K. Jain, and R. Pamula. **An efficient use of ensemble methods to predict students academic performance**.*4th International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad. IEEE, pp. 1-6, 2018.doi: 10.1109/RAIT.2018.8389056
11. A.Almasri, E.Celebi, and R. S. Alkhawaldeh.**EMT: Ensemble Meta-Based Tree Model for Predicting Student Performance**. *Scientific Programming*, vol. 2019, Article ID 3610248, 13 pages, 2019.
12. M. Pandey, and S. Taruna.**A Comparative Study of Ensemble Methods for Students' Performance Modeling**. *International Journal of Computer Applications*,vol. 103,no.8, pp.26-32, Oct 2014. https://doi.org/10.5120/18095-9151
13. D. Opitz, and R. Maclin.**Popular Ensemble Methods: An Empirical Study**. *Journal of Artificial Intelligence Research*, vol. 11,pp. 169-198, 1999.
14. C. Beaulac, and J. S. Rosenthal.**Predicting University Students' Academic Success and Major Using Random Forests**. *Research in Higher Education*, vol. 60, no. 7, pp. 1048-1064, Nov 2019.
15. E. Fernandes et al. **Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil**. *Journal of Business Research*, vol. 94, pp. 335-343, Jan 2019.
16. P. Kamal, and S. Ahuja. **An ensemble-based model for prediction of academic performance of students in undergrad professional course**. *Journal of Engineering, Design and Technology*, vol. 17 no. 4, pp. 769-781, 2019. https://doi.org/10.1108/JEDT-11-2018-0204
17. A. J. Wyner, M. Olson, and J. Bleich. **Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers**. *Journal of Machine Learning Research*, vol. 18, pp. 1-33, 2017.
18. U. D. Kumar, **Business Analytics- The Science of Data Driven Decision Making**. New Delhi: Wiley India, 2017, pp. 403-405.
19. R. E. Schapire. **The Boosting Approach to Machine Learning: An Overview**. In: Denison D.D., Hansen M.H., Holmes C.C., Mallick B., Yu B. (eds) Nonlinear Estimation and Classification. Lecture Notes in Statistics, Springer, New York, NY,vol 171,pp 149-171, 2003.
20. **The R project for statistical computing**. https://www.r-project.org/
21. J.E. Holden, W. H. Finch, andK. Kelley. **A Comparison of Two-Group Classification Methods, Educational and Psychological Measurement**. 71(5). Sage, 2011, pp. 870–901.
22. **Package rpart**. https://cran.r-project.org/web/packages/rpart/rpart.pdf https://doi.org/10.1145/1882471.1882479
23. G. Forman, andM. Scholz.**Apples to apples in cross-validation studies: Pitfalls in classifier performance measurement**. *ACM SIGKDD Explorations*, vol.12(1), pp. 49–57, 2010.
24. **Package adabag.** https://cran.r-project.org/web/packages/adabag/adabag.pdf
25. **Package gbm**. https://cran.r-project.org/web/packages/gbm/gbm.pdf
26. **Package xgboost**. https://cran.r-project.org/web/packages/xgboost/xgboost.pdf
27. H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun and V. Vapnik, **Boosting and Other Ensemble Methods**. *Neural Computation*, vol. 6, no. 6, pp. 1289-1301, Nov. 1994. https://doi.org/10.1162/neco.1994.6.6.1289