Improving Classification of Fraudulent Sales

Barry E. King
Butler University – Lacy School of Business
4600 Sunset Avenue
Indianapolis, Indiana USA

Abstract

This article presents an improved solution to classifying fraudulent sales. An original k-nearest neighbor solution for a dataset of more than fifteen thousand cases yielded a misclassification rate of 0.058 where eight percent of the observations were fraudulent. An improved solution using a boosted C5.0 algorithm yielded a misclassification rate of 0.038. The solution was expanded to recognize that false positives (classifying a fraudulent sale as clean) were five times as costly as were false negatives (classifying a clean sale as fraudulent). The misclassification rate for this expanded solution was 0.058 but lowered the misclassification cost by twenty-one percent.

Keywords - *binary classification, machine learning, k-nearest neighbor, C5.0 algorithm.*

I. INTRODUCTION

In June 2016, JP Murillo [1] wrote a blog entry classifying sales cases as fraudulent or okay. He used k-nearest neighbor to develop his classification. His misclassification rate was 0.058. Here we seek to improve the misclassification rate by using a boosted C5.0 algorithm and then extend the solution to consider that a false positive (classifying a fraudulent sale as okay) is costlier to the firm than is a false negative (classifying an okay sale as fraudulent).

II. THE PROBLEM

Murillo reported 401,146 sales in a dataset. 14,462 were clean sales, 1,270 were fraudulent, and the remaining had an unknown status. After eliminating cases with missing values, we obtained 15,546 usable observations of which 1,199 were fraudulent and the remainder were clean.

Table I reports five usable attributes of the dataset.

TABLE I

Five Usable Attributes of the Dataset

Attribute	Description
ID	Salesperson identification number
Val	Sales dollar amount
Prod	Product id
Quant	Quantity
Insp	Fraud inspection status (the target variable)

Murillo used k-nearest neighbor to classify sales as "fraud" or "ok." He obtained a 0.058 misclassification rate. See Table II.

TABLE II

k-Nearest Neighbor Confusion Matrix

Predicted	Actual	
	fraud	**ok**
fraud	131 0.042	108 0.035
ok	73 0.023	2796 0.900

Note. First table cell entry is N.
Second entry is N / table total.

III. CLASSIFICATION

A. Boosted c5.0 Model

Variables ID and Prod have 2,821 levels and 798 levels respectively. Although tree-based algorithms do not require factors to be represented by indicator or dummy variables, we found creating such variables was useful.

The dataset was partitioned into 75 percent train (n = 11,661) and 25 percent test (n = 3885).

The C5.0 algorithm was boosted with ten trials.

The results of this run are shown in Table III.

TABLE III

Boosted C5.0 Model Confusion Matrix

Predicted	Actual	
	fraud	**ok**
fraud	194 0.050	41 0.011
ok	105 0.027	3545 0.912

Note: Misclassification rate = 0.038.

The boosted C5.0 misclassification rate of 0.038 is a considerable improvement to the k-nearest neighbor misclassification rate of 0.058.

IV. MODEL WITH A COST MATRIX

Using B. Lantz's technique of incorporating asymmetric costs into a C5.0 model [2], we expanded the model to include a cost of five for predicting a fraudulent sale as okay and a cost of one for predicting an okay sale as fraudulent. See Table IV.

TABLE IV

Asymmetric Cost Matrix

Predicted	Actual	
	fraud	ok
fraud	0	1
ok	5	0

The results of this run are shown in Table V.

TABLE V

C5.0 Confusion Matrix When Costs Are Applied

Predicted	Actual	
	fraud	ok
fraud	244 0.063	170 0.044
ok	55 0.014	3416 0.879

Note: Misclassification rate = 0.058.
Calculated cost = 445.

The misclassification rate for the cost enhanced model is 0.058, worse than the misclassification rate of 0.038 for the boosted C5.0 without a cost matrix. This is expected. The solution will worsen the more you constrain a problem. However, the calculated cost for the boosted C5.0 model is 566 while that for the cost enhanced model is 445. Here we traded accuracy for lower misclassification costs.

V. CONCLUSION

Selecting cases at random from the dataset would produce about eight percent fraudulent cases, the fraudulent proportion of the dataset. Using the boosted C5.0 algorithm with 0.038 misclassification rate, improves the prediction accuracy. See Fig. 1.

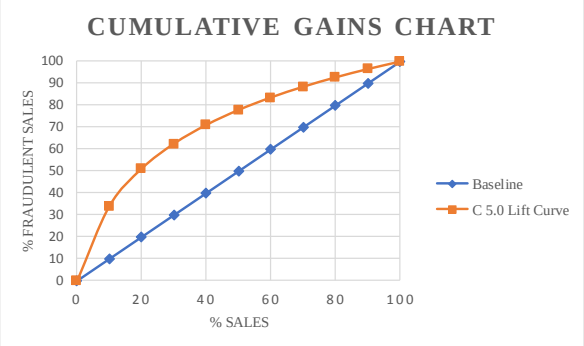

Fig.1: Boosted C5.0 model lift chart

REFERENCES

[1] Murillo, J. P. (2016). Predicting fraudulent sales. [Online] https://rpubs.com/jpmurillo/fraudulentsales.
[2] Lantz, B. (2015). Machine Learning with R, 2nd edition. Birmingham, UK: Packt.