Applying Machine Learning to Bank Churn Prediction

Rajbir Bhattacharjee

**Abstract** Regarding *Lorem Ipsum* and according to Wikipedia, in publishing and graphic design, *lorem ipsum* is common placeholder text used to demonstrate the graphic elements of a document or visual presentation such as font, typography, and layout. Although it resembles Latin, it does not have any inherent meaning. Publishers often use *lorem ipsum* when displaying typeface to direct focus to the presentation. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam sapien. Proin euismod metus id elit. Aliquam posuere orci nec lacus. Vivamus consectetuer, turpis non vulputate faucibus, nisl lectus pretium urna, at mollis turpis arcu non quam.

1 Introduction

The data-set considered in this paper is labelled sample of a bank’s customer data. The data is labeled whether the customer left the bank or not. The task is a classification task to predict whether a customer will leave the bank or not.

The dataset has the following fields:

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Notes** |
| RowNumber | Integer | Row Number |
| CustomerId | Integer | Customer Id, no specific pattern noted |
| Surname | String | Surname of the customer |
| CreditScore | Integer | Credit score of the customer |
| Geography | String | Country of the customer, has 3 distinct values, France, Germany and Spain |
| Gender | String | Male or Female |
| Age | Integer | Age of the customer |
| Tenure | Integer | Number of years with the bank |
| Balance | Float | Account balance |
| NumOfProducts | Integer | Number of products customer is subscribed to |
| HasCrCard | Integer | Denotes whether a customer has a credit card or not |
| IsActiveMember | Integer | Denotes whether customer is active or not |
| Estimated Salary | Float | Estimated salary of the customer |
| Exited | Integer | Target class label: 0/1 indicating whether the customer exited the bank or not. |

The data set has 10000 examples with no missing values. However, the data set suffers from moderate class imbalance with about 25% of the samples have the label Exited=1, and the remaining 75% have the label Exited=0.

The motivation to build a machine learning model is to be able to predict which customers will exit the bank after training on the data. A further objective of the study is to compare how different models perform, along with techniques to remedy data imbalance.

2 Research

This work compared the performance various several machine learning models to the dataset. A further goal of this work is to research techniques to remedy data imbalance and compare how they perform on the dataset.

In a classification problem, the ideal situation is where each target label has equal probability of occurrence in the real world, as well as the sample data-set. However, that is seldom the case, and in most real-world problems, the class labels are highly skewed. Consider the case of predicting cancer – only a small percentage of the population have cancer, and the vast majority of the population do not have cancer. Further imbalance maybe introduced by the sampling methodology.

If we are predicting two classes, and the chances of a sample being a +ve is 1%, and the chance of a sample being a -ve is 99%, a machine learning model can achieve an accuracy of 99% by simply predicting -ve all the time. However, such a prediction is of little use. Hence accuracy is not a good judge of the quality of prediction. Better measures of judging accuracy are measures like precision, recall and F-1 score. The F-1 score is particularly interesting because it uses just one number to indicate both false positives and false negatives. Another metric that is frequently used is the confusion matrix.[[1]](#footnote-1)

The goal of any model is to be able to accurately predict both +ve and -ve classes. However, since in real world phenomena, -ve examples vastly outnumber +ve examples, the model has very little opportunity and incentive to learn how to predict the minority class correctly. This generally results in a very good performance for predicting the majority class, but a very bad performance for the minority class. For such types of problems, the challenges introduced by data imbalance must be remedied.

TODO: Discuss classifiers

One way to deal with data imbalance the use of alternate cut-offs in classifiers that output each prediction with a probability. This probability can be adjusted so that the minority class is favored. The use of alternate cut-offs does not result in any change in the model.[1]

Another method of dealing with imbalance is to adjust the prior probabilities of the classes so that it results in an increase in the probability of the minority class being predicted. Such a method can help improve the performance of a Bayesian learner. Weiss and Provost (2001)[2] suggest that using prior probabilities that describe the true occurrence of a class in the real-world biases the model towards predicting the majority class, and using a more balanced prior probability is likely to improve the performance of the model.

Yet another way to deal with imbalance is to use unequal case weights. Ting (2002)[3] describes one method in which they adjust the weight of a sample to reflect the cost incurred in misclassifying it, with the minority samples having a greater weight than the majority class.

A similar effect also happens in boosting, where the probability distribution over all the examples is changed at each iteration, which affects the sampling, and results in more samples from the misclassified instances being drawn. Since the minority class is likely to be misclassified more initially, it is likely that later samples will contain more instances of the minority class, and the overall classifier will learn to classify the minority class correctly.[4]

Cost-sensitive training has been the subject of much inquiry. In such methods, the cost function modified such misclassification of an instance belonging to the minority class incurs a much heavier penalty than misclassification of the majority class. The modified cost function results in a change in the model so that the model learns to classify the minority class better. This contrasts with the use of alternate cut-offs where there was no change in the model as such. Another variation of alternate cut-offs is one which not only assigns different costs to misclassification of the majority or minority class, but also allows for different costs for certain types of errors.

Many classification tree models including CART can be modified to incorporate different costs for different classes. The cost can take into account the cost of the mistake, the probability of making a mistake, and the prior probability of the class for a sophisticated.[5]

3 Methodology

4 Evaluation

5 Conclusion

References

1. Kuhn, M., Johnson, K., Applied Predictive Modeling, Springer, 2013
2. Weiss, G.M., Provost, F.: The effect of class distrubition on classifier learning. Technical report ML-TR 43, Department of Computer Scinece, Rutgers University, 2001
3. Ting, K.M., An instance weighting method to induce cost-sensitive trees. IEEE Transactions on knowledge and data engineering, Vol. 14, No. 3, 2002
4. Schapire, E., Freund, Y.: Experiments with a new boosting algorithm. Machine Learning: Proceedings of the Thirteenth International Conference, 1996.
5. Johnson, R.A. and Wichern D.W., Applied Multivariate StatisticalAnalysis, Pearson, 2007

1. In this discussion, we will talk about the case of a binary classification for simplicity. However, the same discussion can generalize to multi-class classification as well. [↑](#footnote-ref-1)