A Logistic Regression Approach to CoIL Challenge 2000

Corey Arnouts[1], Adam Douglas[1], Jason Givens-Doyle[1], & Michael Silva[1]

[1] MS in Data Science Students CUNY School of Professional Studies

Author Note

Correspondence concerning this article should be addressed to Corey Arnouts, 119 W 31st St., New York, NY 10001. E-mail: Corey.Arnouts@spsmail.cuny.edu

Abstract

A logistic regression based solution to the CoIL Challenge 2000 is described. The challenge consists of correctly identifying potential customers for an insurance product, and describing their characteristics.

Keywords: CoIL Challenge, Logistic Regression

Word count: X

A Logistic Regression Approach to CoIL Challenge 2000

Introduction

Businesses use data science to extract insights from data. It has many practical business applications. Identifying households to include in a marketing campaign is one application. One example using real world data is the Computational Intelligence and Learning (CoIL) Challenge. The CoIL Challenge competition was held from March 17 to May 8 in 2000. The challenge is to:

1. Identify potential customers for an insurance policy; and
2. Provide a description of this customer base.

In total 147 participants registered and 43 submitted solutions (Putten, Ruiter, & Someren, 2000). In this paper we set out to complete the first part of the COIL Challenge. **SUMARISE FINDINGS?**

Literature Review

Participants used a variety of approaches in formulating their submissions including:

- Classification Trees with Bagging (White & Liu, 2000);
- C4.5 (Seewald, 2000);
- Evolutionary Algorithm (Koudijs, 2000);
- Inductive Learning by Logic Minimization (ILLM) (Gamberger, 2000);
- Mask Perceptron with Boosting (Leckie & Ferra, 2000);
- Naïve Bayes (Elkan, 2000); and

- Neural Networks(Crocoll, 2000), (Kim & Street, 2000);
- Scoring System (Lewandowski, 2000);
- Support Vector Machines; and
- XCS (Greenyer, 2000).

The maximum number of policyowners that could be found was 238. The winning model (Elkan, 2000) selected 121 policy owners. Random selection results in 42 policy owners. The standard benchmark tests result in 94 (k-nearest neighbor), 102 (naïve bayes), 105 (neural networks) and 118 (linear!) policy owners. (Putten et al., 2000)

Methodology

Experimentation and Results

Discussion and Conclusions

References

Crocoll, W. M. (2000). Artificial neural network portion of coil study. Retrieved from http://www.liacs.nl/~putten/library/cc2000/CROCOL~1.pdf

Elkan, C. (2000). CoIL challenge 2000 entry. Retrieved from http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/ELKANP~1.pdf

Gamberger, D. (2000). Solution based on illm confirmation rule. Retrieved from http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/GAMBER~1.pdf

Greenyer, A. (2000). Coil 2000 competition. The use of a learning classifier system jxcs. Retrieved from http://www.liacs.nl/~putten/library/cc2000/GREENY~1.pdf

Kim, Y., & Street, W. N. (2000). CoIL challenge 2000: Choosing and explaining likely caravan insurance customers. Retrieved from http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/STREET~1.pdf

Koudijs, A. (2000). CoIL challenge 2000 submission for the description task. Retrieved from http://www.liacs.nl/~putten/library/cc2000/KOUDIJ~1.pdf

Leckie, C., & Ferra, H. (2000). COIL challenge 2000 description task. Retrieved from http://www.liacs.nl/~putten/library/cc2000/LECKIE~1.pdf

Lewandowski, A. (2000). How to detect potential customers. Retrieved from http://www.liacs.nl/~putten/library/cc2000/LEWAND~1.pdf

Putten, P., Ruiter, M., & Someren, M. (2000). CoIL challenge 2000 tasks and results: Predicting and explaining caravan policy ownership.

Seewald, A. (2000). CoIL challenge 2000 submitted solution. Retrieved from http://www.liacs.nl/~putten/library/cc2000/SEEWAL~1.pdf

White, A. P., & Liu, W. Z. (2000). The coil challenge: An application of classification trees with bootstrap aggregation. Retrieved from http://www.liacs.nl/~putten/library/cc2000/WHITEP~1.pdf

Appendices

- Supplemental tables and/or figures.

R statistical programming code.