A Logistic Regression Approach to CoIL Challenge 2000

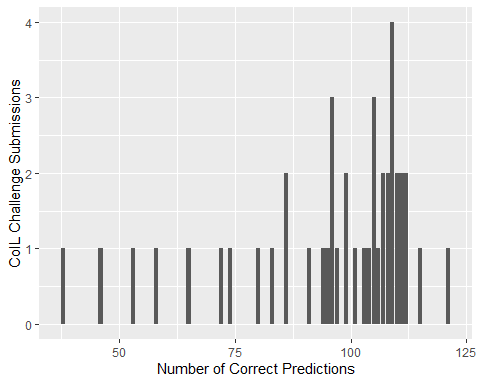
Corey Arnouts, Adam Douglas, Jason Givens-Doyle, Michael Silva

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.

# 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, and Someren 2000). The maximum number of policyowners that could be found was 238. The submissions identified 95 policy owners on average. The winning model (Elkan 2000) identified 121 policy owners. Random selection results in identifying 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, Ruiter, and Someren 2000). In this paper we set out to complete the first part of the COIL Challenge.

# Literature Review

Participants used a variety of approaches in formulating their submissions including: Boosted Decision Tree (McKone and Stenger 2000), Classification and Regression Tree (CART) (Simmonds 2000), Classification Trees with Bagging (White and Liu 2000), C4.5 (Rickets 2000; Seewald 2000), Evolutionary Algorithm (Koudijs 2000), Fuzzy Classifier (János Abonyi 2000; Kaymak and Setnes 2000), Genetic Algorithms and Hill-climbers (Carter 2000), Inductive Learning by Logic Minimization (ILLM) (Gamberger 2000; Šmuc 2000), Instance Based Reasoning (iBARET) (Mikšovský and Klema 2000), K-Means (Vesanto and Sinkkonen 2000), KXEN (Bera and Lamy 2000), LOGIT (Doornik and Moyle 2000), Mask Perceptron with Boosting (Leckie and Ferra 2000), Midos Algorithm (Krogel 2000), N-Tuple Classifier (Jorgensen and Linneberg 2000), Naïve Bayes (Elkan 2000; Kontkanen 2000), Neural Networks(Brierley 2000; Crocoll 2000; Kim and Street 2000; Shtovba and Mashnitskiy 2000), Phase Intervals and Genetic Algorithms (Shtovba 2000), Scoring System (Lewandowski 2000), Support Vector Machines(Keerthi and Ong 2000), and XCS (Greenyer 2000).

# Methodology

# Experimentation and Results

# Discussion and Conclusions

# References

Bera, Michel, and Bertrand Lamy. 2000. “Kxen at Coil Challenge 2000.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/BERAPS~1.pdf>.

Brierley, Philip. 2000. “COIL 2000 Challenge: Characteristics of Caravan Insurance Policy Owners.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/BRIERL~1.pdf>.

Carter, Jonathan. 2000. “Coil 2000 Challenge Submission: Genetic Algorithms and Hill-Climbers.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/CARTER~1.pdf>.

Crocoll, William M. 2000. “Artificial Neural Network Portion of Coil Study.” <http://www.liacs.nl/~putten/library/cc2000/CROCOL~1.pdf>.

Doornik, Jurgen A., and Steve Moyle. 2000. “LOGIT Modelling.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/MOYLEP~1.pdf>.

Elkan, Charles. 2000. “CoIL Challenge 2000 Entry.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/ELKANP~1.pdf>.

Gamberger, Dragan. 2000. “Solution Based on Illm Confirmation Rule.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/GAMBER~1.pdf>.

Greenyer, Andrew. 2000. “Coil 2000 Competition. The Use of a Learning Classifier System Jxcs.” <http://www.liacs.nl/~putten/library/cc2000/GREENY~1.pdf>.

János Abonyi, Hans Roubos. 2000. “A Simple Fuzzy Classifier Based on Inconsistency Analysis of Labeled Data.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/ABONYI~1.pdf>.

Jorgensen, Thomas Martini, and Christian Linneberg. 2000. “Subspace Projections – an Approach to Variable Selection and Modeling.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/JORGEN~1.pdf>.

Kaymak, Uzay, and Magne Setnes. 2000. “Target Selection Based on Fuzzy Clustering: A Volume Prototype Approach to Coil Challenge 2000.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/KAYMAK~1.pdf>.

Keerthi, S. Sathiya, and Chong Jin Ong. 2000. “Solution of the Coil Challenge 2000 Task Using Support Vector Machines.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/KEERTH~1.pdf>.

Kim, YongSeog, and W. Nick Street. 2000. “CoIL Challenge 2000: Choosing and Explaining Likely Caravan Insurance Customers.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/STREET~1.pdf>.

Kontkanen, Petri. 2000. “CoIL 2000 Submission.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/KONTKA~1.pdf>.

Koudijs, Arnold. 2000. “CoIL Challenge 2000 Submission for the Description Task.” <http://www.liacs.nl/~putten/library/cc2000/KOUDIJ~1.pdf>.

Krogel, Mark-André. 2000. “A Data Mining Case Study.” <http://www.liacs.nl/~putten/library/cc2000/KROGEL~1.pdf>.

Leckie, Chris, and Herman Ferra. 2000. “COIL Challenge 2000 Description Task.” <http://www.liacs.nl/~putten/library/cc2000/LECKIE~1.pdf>.

Lewandowski, Achim. 2000. “How to Detect Potential Customers.” <http://www.liacs.nl/~putten/library/cc2000/LEWAND~1.pdf>.

McKone, Tom, and Curt Stenger. 2000. “COIL Challenge 2000 Submission.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/MCKONE~1.pdf>.

Mikšovský, Petr, and Jirí Klema. 2000. “CoIL Challenge 2000.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/MIKSOV~1.pdf>.

Putten, Peter, Michel Ruiter, and Maarten Someren. 2000. “CoIL Challenge 2000 Tasks and Results: Predicting and Explaining Caravan Policy Ownership.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/PUTTEN~1.pdf>.

Rickets, Paul. 2000. “CoIL Challenge 2000 Submission.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/RICKET~1.pdf>.

Seewald, Alexander. 2000. “CoIL Challenge 2000 Submitted Solution.” <http://www.liacs.nl/~putten/library/cc2000/SEEWAL~1.pdf>.

Shtovba, Serhiy. 2000. “Phase Intervals and Genetic Algorithms Based Competition Task Solution.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/SHTOVB~2.pdf>.

Shtovba, Serhiy, and Yakiv Mashnitskiy. 2000. “The Backpropagation Multilayer Feedforward Neural Network Based Competition Task Solution.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/SHTOVB~1.pdf>.

Simmonds, Robert M. 2000. “ACT Study Report Using Classification and Regression Tree (Cart) Analysis.” <http://www.liacs.nl/~putten/library/cc2000/SIMMON~1.pdf>.

Šmuc, Tomislav. 2000. “COIL 2000 Challenge Solution Based on Illm-Sg Methodology.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/SMUCPS~1.pdf>.

Vesanto, Juha, and Janne Sinkkonen. 2000. “Submission for the Coil Challenge 2000.” <http://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/VESANT~1.pdf>.

White, A. P., and W. Z. Liu. 2000. “The Coil Challenge: An Application of Classification Trees with Bootstrap Aggregation.” <http://www.liacs.nl/~putten/library/cc2000/WHITEP~1.pdf>.

# Appendices

## Data Dictionary

|  |  |
| --- | --- |
| Name | Description |
| MOSTYPE | Customer Subtype |
| MAANTHUI | Number of houses |
| MGEMOMV | Avg size household |
| MGEMLEEF | Avg age |
| MOSHOOFD | Customer main type |
| MGODRK | Roman catholic |
| MGODPR | Protestant |
| MGODOV | Other religion |
| MGODGE | No religion |
| MRELGE | Married |
| MRELSA | Living together |
| MRELOV | Other relation |
| MFALLEEN | Singles |
| MFGEKIND | Household without children |
| MFWEKIND | Household with children |
| MOPLHOOG | High level education |
| MOPLMIDD | Medium level education |
| MOPLLAAG | Lower level education |
| MBERHOOG | High status |
| MBERZELF | Entrepreneur |
| MBERBOER | Farmer |
| MBERMIDD | Middle management |
| MBERARBG | Skilled labourers |
| MBERARBO | Unskilled labourers |
| MSKA | Social class A |
| MSKB1 | Social class B1 |
| MSKB2 | Social class B2 |
| MSKC | Social class C |
| MSKD | Social class D |
| MHHUUR | Rented house |
| MHKOOP | Home owners |
| MAUT1 | 1 car |
| MAUT2 | 2 cars |
| MAUT0 | No car |
| MZFONDS | National Health Service |
| MZPART | Private health insurance |
| MINKM30 | Income < 30.000 |
| MINK3045 | Income 30-45.000 |
| MINK4575 | Income 45-75.000 |
| MINK7512 | Income 75-122.000 |
| MINK123M | Income >123.000 |
| MINKGEM | Average income |
| MKOOPKLA | Purchasing power class |
| PWAPART | Contribution private third party insurance |
| PWABEDR | Contribution third party insurance (firms) |
| PWAAND | Contribution third party insurane (agriculture) |
| PPERSAUT | Contribution car policies |
| PBESAUT | Contribution delivery van policies |
| PMOTSCO | Contribution motorcycle/scooter policies |
| PVRAAUT | Contribution lorry policies |
| PAANHANG | Contribution trailer policies |
| PTRACTOR | Contribution tractor policies |
| PWERKT | Contribution agricultural machines policies |
| PBROM | Contribution moped policies |
| PLEVEN | Contribution life insurances |
| PPERSONG | Contribution private accident insurance policies |
| PGEZONG | Contribution family accidents insurance policies |
| PWAOREG | Contribution disability insurance policies |
| PBRAND | Contribution fire policies |
| PZEILPL | Contribution surfboard policies |
| PPLEZIER | Contribution boat policies |
| PFIETS | Contribution bicycle policies |
| PINBOED | Contribution property insurance policies |
| PBYSTAND | Contribution social security insurance policies |
| AWAPART | Number of private third party insurance |
| AWABEDR | Number of third party insurance (firms) |
| AWALAND | Number of third party insurane (agriculture) |
| APERSAUT | Number of car policies |
| ABESAUT | Number of delivery van policies |
| AMOTSCO | Number of motorcycle/scooter policies |
| AVRAAUT | Number of lorry policies |
| AAANHANG | Number of trailer policies |
| ATRACTOR | Number of tractor policies |
| AWERKT | Number of agricultural machines policies |
| ABROM | Number of moped policies |
| ALEVEN | Number of life insurances |
| APERSONG | Number of private accident insurance policies |
| AGEZONG | Number of family accidents insurance policies |
| AWAOREG | Number of disability insurance policies |
| ABRAND | Number of fire policies |
| AZEILPL | Number of surfboard policies |
| APLEZIER | Number of boat policies |
| AFIETS | Number of bicycle policies |
| AINBOED | Number of property insurance policies |
| ABYSTAND | Number of social security insurance policies |
| CARAVAN | Number of mobile home policy |

## R statistical programming code.

# CoIL Challenge Source Code   
library(tidyverse)   
library(caret)   
   
## Download the data sets from UCI if they are not present   
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/tic-mld/"   
files <- c("ticdata2000.txt", "ticeval2000.txt", "tictgts2000.txt")   
for (file\_name in files) {   
 file\_path <- paste0("data/", file\_name)   
 file\_url <- paste0(url, file\_name)   
 if (!file.exists(file\_path)) {   
 message(paste("Downloading", file\_name))   
 download.file(file\_url, file\_path)   
 }   
}   
   
## Read in and clean the data   
prepare\_data <- function(df){   
 names(df) <- c(   
 "MOSTYPE", "MAANTHUI", "MGEMOMV", "MGEMLEEF", "MOSHOOFD", "MGODRK",   
 "MGODPR", "MGODOV", "MGODGE", "MRELGE", "MRELSA", "MRELOV", "MFALLEEN",   
 "MFGEKIND", "MFWEKIND", "MOPLHOOG", "MOPLMIDD", "MOPLLAAG", "MBERHOOG",   
 "MBERZELF", "MBERBOER", "MBERMIDD", "MBERARBG", "MBERARBO", "MSKA",   
 "MSKB1", "MSKB2", "MSKC", "MSKD", "MHHUUR", "MHKOOP", "MAUT1", "MAUT2",   
 "MAUT0", "MZFONDS", "MZPART", "MINKM30", "MINK3045", "MINK4575",   
 "MINK7512", "MINK123M", "MINKGEM", "MKOOPKLA", "PWAPART", "PWABEDR",   
 "PWALAND", "PPERSAUT", "PBESAUT", "PMOTSCO", "PVRAAUT", "PAANHANG",   
 "PTRACTOR", "PWERKT", "PBROM", "PLEVEN", "PPERSONG", "PGEZONG",   
 "PWAOREG", "PBRAND", "PZEILPL", "PPLEZIER", "PFIETS", "PINBOED",   
 "PBYSTAND", "AWAPART", "AWABEDR", "AWALAND", "APERSAUT", "ABESAUT",   
 "AMOTSCO", "AVRAAUT", "AAANHANG", "ATRACTOR", "AWERKT", "ABROM",   
 "ALEVEN", "APERSONG", "AGEZONG", "AWAOREG", "ABRAND", "AZEILPL",   
 "APLEZIER", "AFIETS", "AINBOED", "ABYSTAND", "CARAVAN")   
   
 MOSTYPE\_labels <- c(   
 "1" = "1 High Income, expensive child",   
 "2" = "2 Very Important Provincials",   
 "3" = "3 High status seniors",   
 "4" = "4 Affluent senior apartments",   
 "5" = "5 Mixed seniors",   
 "6" = "6 Career and childcare",   
 "7" = "7 Dinki's (double income no kids)",   
 "8" = "8 Middle class families",   
 "9" = "9 Modern, complete families",   
 "10" = "10 Stable family",   
 "11" = "11 Family starters",   
 "12" = "12 Affluent young families",   
 "13" = "13 Young all american family",   
 "14" = "14 Junior cosmopolitan",   
 "15" = "15 Senior cosmopolitans",   
 "16" = "16 Students in apartments",   
 "17" = "17 Fresh masters in the city",   
 "18" = "18 Single youth",   
 "19" = "19 Suburban youth",   
 "20" = "20 Ethnically diverse",   
 "21" = "21 Young urban have-nots",   
 "22" = "22 Mixed apartment dwellers",   
 "23" = "23 Young and rising",   
 "24" = "24 Young, low educated",   
 "25" = "25 Young seniors in the city",   
 "26" = "26 Own home elderly",   
 "27" = "27 Seniors in apartments",   
 "28" = "28 Residential elderly",   
 "29" = "29 Porchless seniors: no front yard",   
 "30" = "30 Religious elderly singles",   
 "31" = "31 Low income catholics",   
 "32" = "32 Mixed seniors",   
 "33" = "33 Lower class large families",   
 "34" = "34 Large family, employed child",   
 "35" = "35 Village families",   
 "36" = "36 Couples with teens 'Married with children'",   
 "37" = "37 Mixed small town dwellers",   
 "38" = "38 Traditional families",   
 "39" = "39 Large religous families",   
 "40" = "40 Large family farms",   
 "41" = "41 Mixed rurals")   
   
 MGEMLEEF\_labels <- c(   
 "1" = "20-30 years",   
 "2" = "30-40 years",   
 "3" = "40-50 years",   
 "4" = "50-60 years",   
 "5" = "60-70 years",   
 "6" = "70-80 years")   
   
 MOSHOOFD\_labels <- c(   
 "1" = "Successful hedonists",   
 "2" = "Driven Growers",   
 "3" = "Average Family",   
 "4" = "Career Loners",   
 "5" = "Living well",   
 "6" = "Cruising Seniors",   
 "7" = "Retired and Religeous",   
 "8" = "Family with grown ups",   
 "9" = "Conservative families",   
 "10" = "Farmers")   
   
 MGODRK\_labels <- c(   
 "0" = "0%",   
 "1" = "1 - 10%",   
 "2" = "11 - 23%",   
 "3" = "24 - 36%",   
 "4" = "37 - 49%",   
 "5" = "50 - 62%",   
 "6" = "63 - 75%",   
 "7" = "76 - 88%",   
 "8" = "89 - 99%",   
 "9" = "100%")   
   
 PWAPART\_labels <- c(   
 "0" = "f 0",   
 "1" = "f 1 â€“ 49",   
 "2" = "f 50 â€“ 99",   
 "3" = "f 100 â€“ 199",   
 "4" = "f 200 â€“ 499",   
 "5" = "f 500 â€“ 999",   
 "6" = "f 1000 â€“ 4999",   
 "7" = "f 5000 â€“ 9999",   
 "8" = "f 10.000 - 19.999",   
 "9" = "f 20.000 - ?")   
   
 set\_to\_1 <- c(12, 8, 6, 7, 1, 2, 36, 3, 37, 10, 20, 38, 11)   
   
 df %>%   
 mutate(CUSTOMER\_TYPE = ifelse(MOSTYPE %in% set\_to\_1, 1, 0)) %>%   
 mutate(CUSTOMER\_TYPE = as.factor(CUSTOMER\_TYPE)) %>%   
 mutate(MOSTYPE = as.factor(MOSTYPE),   
 MGEMLEEF = as.factor(MGEMLEEF),   
 MOSHOOFD = as.factor(MOSHOOFD),   
 MGODRK = as.factor(MGODRK),   
 PWAPART = as.factor(PWAPART),   
 CARAVAN = as.factor(CARAVAN)) %>%   
 mutate(MOSTYPE = recode(MOSTYPE, !!!MOSTYPE\_labels),   
 MGEMLEEF = recode(MGEMLEEF, !!!MGEMLEEF\_labels),   
 MOSHOOFD = recode(MOSHOOFD, !!!MOSHOOFD\_labels),   
 MGODRK = recode(MGODRK, !!!MGODRK\_labels),   
 PWAPART = recode(PWAPART, !!!PWAPART\_labels))   
}   
   
eval <- read.delim("data/ticeval2000.txt", header = FALSE)   
temp <- read.delim("data/tictgts2000.txt", header = FALSE)   
eval$CARAVAN <- temp$V1   
eval <- prepare\_data(eval)   
df <- prepare\_data(read.delim("data/ticdata2000.txt", header = FALSE))   
   
## Create the train and test sets   
set.seed(42)   
train\_index <- createDataPartition(df$CARAVAN, p = .7, list = FALSE)   
train <- df[train\_index,]   
test <- df[-train\_index,]   
   
## Correct the data imbalance through over sampling   
up\_train <- upSample(x = select(train, -CARAVAN),   
 y = train$CARAVAN,   
 yname = "CARAVAN")   
   
# ## Looking for important variables   
# set.seed(42)   
# library(randomForest)   
# rf\_fit <- randomForest(CARAVAN ~ ., up\_train)   
# varImpPlot(rf\_fit)   
#   
# ## Find important customer types   
# crosstab <- up\_train %>%   
# select(CARAVAN, MOSTYPE) %>%   
# table() %>%   
# data.frame()   
#   
# crosstab %>%   
# group\_by(MOSTYPE) %>%   
# summarise(total = sum(Freq)) %>%   
# merge(crosstab) %>%   
# mutate(share = Freq / total) %>%   
# filter(CARAVAN == 1, share > 0.5) %>%   
# arrange(desc(share)) %>%   
# select(MOSTYPE, share)   
   
## Model Building   
   
### Model 1 - Top 5 Important Variables from Random Forest   
model1 <- glm(CARAVAN ~ MOSTYPE + PPERSAUT + MOSHOOFD + PBRAND + APERSAUT,   
 family = binomial(link = "logit"),   
 up\_train)   
   
### Model 2 - Special Customer Type Groups   
model2 <- glm(CARAVAN ~ CUSTOMER\_TYPE + PPERSAUT + APERSAUT,   
 family = binomial(link = "logit"),   
 up\_train)   
   
## Model Evaluation   
   
score\_model <- function(model, data, threshold = 0.5, predictions = FALSE){   
 ## Provides model scoring data   
 #   
 # INPUTS   
 #   
 # model = logit model object   
 # data = data frame to make predictions for   
 # threshold (optional) = the cutpoint to assign a 1 or 0 response   
 # predictions (optional) = 1 or 0 you want to use for the predicitions   
 #   
 # RETURNS (list)   
 #   
 # cm = Confusion Matrix output from caret   
 # correct = the number of correct CARAVAN = 1 predictions   
 # accuracy = the accuracy of the CARAVAN = 1 predictions   
   
 # Generate the predicted outcome   
 if(!predictions){   
 glm\_predictions <- predict.glm(model, data, "response")   
 predictions <- ifelse(glm\_predictions >= threshold, 1, 0)   
 }   
 data$yhat <- predictions   
   
 # Generate a confusion matrix   
 cm <- confusionMatrix(factor(predictions), factor(data$CARAVAN))   
   
 # Get the number of correct CARAVAN = 1 Predictions   
 correct <- data %>%   
 filter(yhat == 1,   
 yhat == CARAVAN) %>%   
 nrow(.)   
   
 # Get the accuracy of the model's CARAVAN = 1 Predictions   
 accuracy <- correct / nrow(data[data$CARAVAN == 1,])   
   
 # Return the data as a list   
 return(list("cm" = cm, "correct" = correct, "accuracy" = accuracy))   
}   
   
results1 <- score\_model(model1, test)   
results2 <- score\_model(model2, test)   
   
## Final Model Accuracy   
score\_model(model1, eval)$correct   
score\_model(model1, eval)$accuracy   
   
score\_model(model2, eval)$correct   
score\_model(model2, eval)$accuracy