A Logistic Regression Approach to CoIL Challenge 2000

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

Abstract

This paper describes a logistic regression based solution to the CoIL Challenge 2000. The challenge consists of correctly identifying potential customers for an insurance product, and describing their characteristics. Models were trained on over sampled data. The model out preformed other's attemps at solving this classification problem.

Key words: CoIL Challenge, Logistic Regression

Introduction

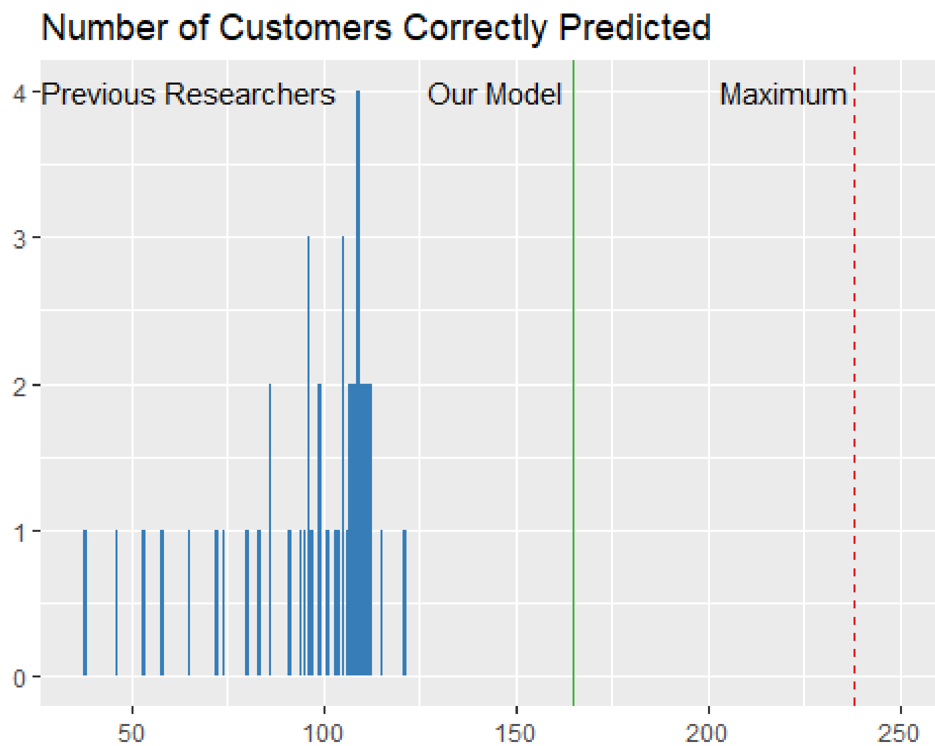
Businesses use data science to extract insights from data. One pratical application is identifying households to include in a marketing campaign. In this paper we set out to identify potential customers for an insurance product using real world data from the Computational Intelligence and Learning (CoIL) Challenge. Specifically we are predicting if a customer is likely candidate for a caravan (mobile home/camper) insurance policy. This is particularly challenging because the data is imballanced (only 348 of the 5,822 records for model training/testing are policy holders).

Literature Review

Fourty-three other research teams have attempted to identify potential insurance policy customers (Putten, Ruiter, and Someren 2000). They used a variety of approaches 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).

The maximum number of potential policy owners that could be found is 238. Previous researchers identified 95 policy owners on average. The best preforming model (Elkan 2000) during the initial challenge identified 121 policy owners. It was a Naïve Bayes, suggesting that probabilities of some of the variables will be useful in identifing potential customers.

A meta analysis of the initial researchers found that simpler algorithms tended to outpreform more complicated ones (Putten, Ruiter, and Someren 2000). With the benefit of these findings, we set out to create a simple logistic regression model that preforms as well or better than the original CoIL Challenge cohort. In the end, our model outpreformed the orignal researcher's model in correctly predicting the customers that would purchase the insurance policy.



Methodology

The CoIL Challenge dataset is composed of 86 variables accross 5,822 observations. An evaluation dataset is provided with 4,000 observations. Five of the predictors are categorical and the remainder are numeric. Most of the predictors have little to no correlation with the variable of interest (CARAVAN).

Experimentation and Results

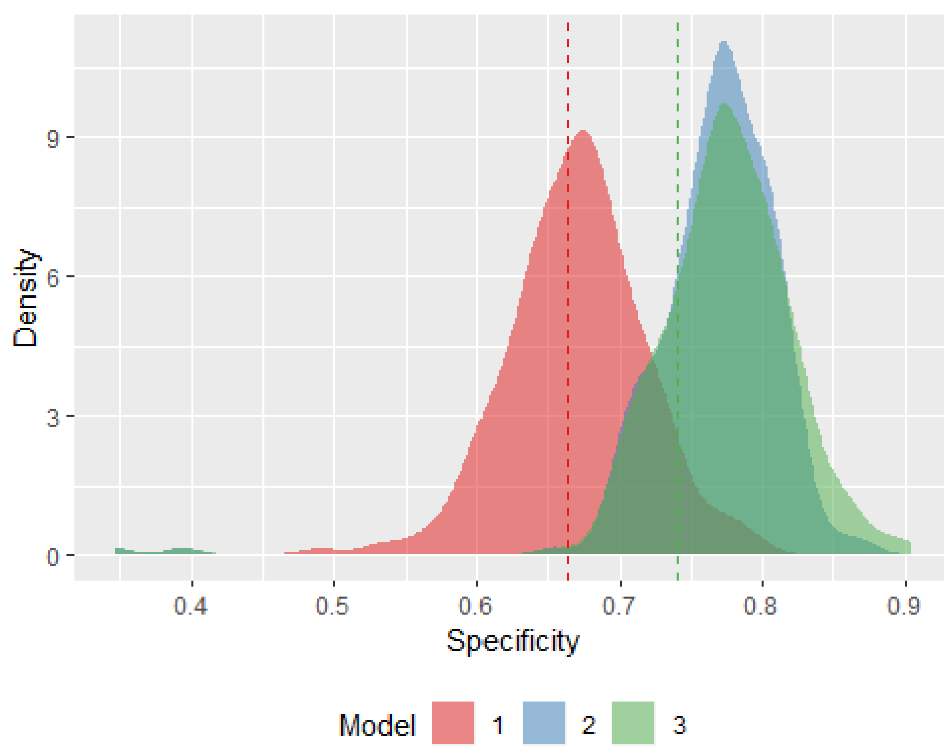
We split the data into training/test sets using a 70/30 split. We corrected the imballance by oversampling the minority class (caravan policy holders). Given the large number of possible predictors we used a random forest to aid in variable selection. A logistic regression model was trained on the oversampled training set using the top five variables selected by the random forest (MOSTYPE, PPERSAUT, MOSHOOFD, PBRAND, APERSAUT).

The MOSTYPE variable has 40 customer types. Not all customer types were statiscally significant predictors. We identified those in the oversampled dataset with a probability greater than 0.5 to purchase the insurance product as the LIKELY_CUSTOMER variable. A figure in the appendix shows all 40 customer types and the probability that they purched the insurance policy in the oversampled dataset.

We fit our second model to the LIKELY_CUSTOMER and PPERSAUT variables. Although it only has two variables, the model preformed well. This seemed in-line with expectations that simple models preform best.

The third model was fit the same as the second model other than the driven growers variable, which is a variable derived from MOSHOOFD variable, specifically it is when the MOSHOOFD variable is equal to 2, which means the specific customer's main type is "Driven Grower" hence the variable name. We chose this because when running decision trees we noticed this specific factor of MOSHOOFD variable often stood out in the decision trees.

In evaluating the models we examined we focused on the specificity. The goal of the CoIL challenge was to accurately predict those would would purchase the insurance policy, so focusing on the model's specificity was the best evaluation metric. In order to get a better sense of how well the model generalizes, we repeatedly retrained and evaluated the model using diffrent samplings of the training dataset. The following figure summarizes the distribution of the specificty the models produced on the test dataset. The dashed line is how our model preformed:



Discussion

Our model outpreforms the original cohort of CoIL Challenge researchers. We found that the being a member of one of the following customer types to be a significant predictor of purchasing the insurance produce (listed in order of the probability of purchasing with the MOSTYPE in parenthesis):

1. Affluent young families (12),
2. Middle class families (8),
3. Career and childcare (6),
4. Double income no kids (7),
5. High Income, expensive child (1),
6. Very Important Provincials (2),
7. Couples with teens 'Married with children' (36),
8. High status seniors (3),
9. Mixed small town dwellers (37),
10. Stable family (10),
11. Ethnically diverse (20),
12. Traditional families (38),
13. Family starters (11)

People at the top of the list (most likely to buy) are generally those who are well to do. This is further reinforced in the model with the inclusion of PPERSAUT or the contribution to car policies. Those with higher contribution levels are more likely to purchase caravan insurance.

"Driven Growers" are made up of the Career and childcare, Double income no kids, and Middle class families customer types. Once again these customer types generally have more disposible income and are likely customers.

These findings are similar to what the winner of the CoIL challenge found (Elkan 2000). They found that a high PPERSAUT (meaning a level 6) or having two car policies to be the strongest single predictors of having caravan insurance. They found the other most statistically significant predictors are:

14. "purchasing power class" is high (5 or higher, especially 7 or higher)
15. a private third party insurance policy
16. a boat policy
17. a social security insurance policy
18. a single fire policy with higher contribution (level 4)

Elkan explain that "Intuitively, these predictors identify customers who have a car and are wealthier than average, and who in general carry more insurance coverage than average. It is not surprising that these are the people who are most likely to have caravan insurance." Our findings are inline with this, although we arrived at this conclusion by examining other variables.

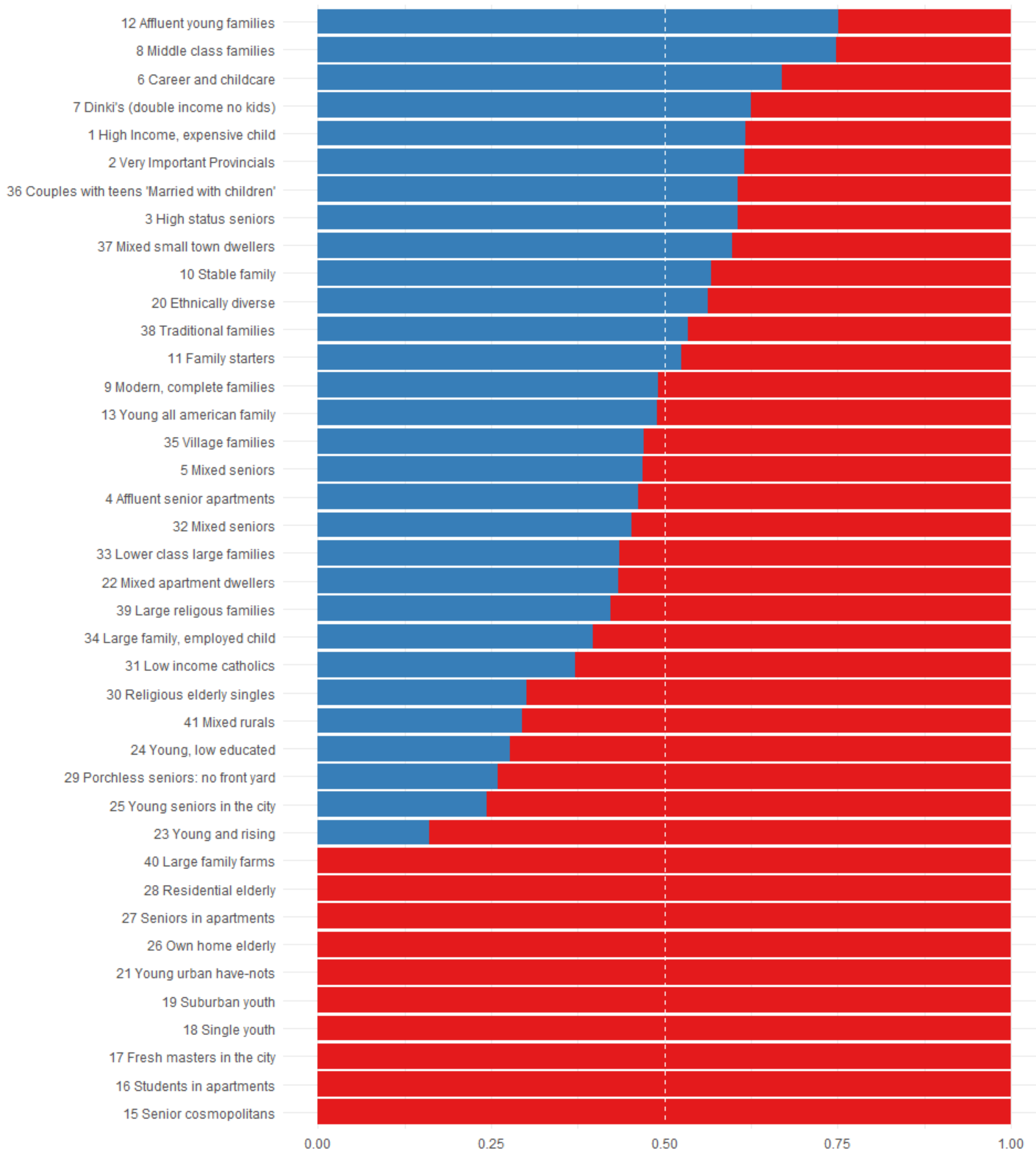
Conclusions

This paper set out to define a model that was effective at identifying potential customers for an insurance policy. We found that a simple logistic regression model with three variables (two which we derived) outpreformed previous research. In general terms the three variables correlate with the wealth of the household. Thus customers that are wealthier than average are most likely candidates to purchase insurance.

Areas for future work include building an ensemble model using the specification of the third model. Our repeated modeling has a distribution where the mean number of correct predictions is 80 which is higher than the 77. It appears that incorporating downsampling into the ensemble might have limited impact on the performance as the mean number of correct predictions is 78.

Appendix

Probability of Purchasing Product (in blue) by Customer Type



Correlation Coefficients for Variables of Interest

	CARAVAN	PPERSAUT	PBRAND	APERSAUT	LIKELY_CUSTOMERS
CARAVAN	1	0.3432	0.1762	0.3188	0.2507
PPERSAUT	0.3432	1	0.1889	0.8879	0.07748
PBRAND	0.1762	0.1889	1	0.2215	0.1657
APERSAUT	0.3188	0.8879	0.2215	1	0.06413
LIKELY_CUSTOMERS	0.2507	0.07748	0.1657	0.06413	1

Data Dictionary for Variables of Interest

Name	Description
CARAVAN	Number of mobile home policy
MOSTYPE	Customer Subtype
MOSHOOFD	Customer main type
PPERSAUT	Contribution car policies
PBRAND	Contribution fire policies
APERSAUT	Number of car policies
LIKELY_CUSTOMERS	MOSTYPE = 12, 8, 6, 7, 1, 2, 36, 3, 37, 10, 20, 38, or 11
DRIVEN_GROWERS	MOSHOOFD = 2

Model Summary

```
Call:
glm(formula = CARAVAN ~ LIKELY_CUSTOMERS + PPERSAUT + DRIVEN_GROWERS,
    family = binomial(link = "logit"), data = up_train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.93776  -1.09454   0.00613   1.15153   1.84219

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.494385   0.052068 -28.701  < 2e-16 ***
LIKELY_CUSTOMERS1  0.908764   0.055316  16.429  < 2e-16 ***
PPERSAUT           0.259265   0.009224  28.109  < 2e-16 ***
DRIVEN_GROWERS1    0.482202   0.085071   5.668 1.44e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 10624.6  on 7663  degrees of freedom
Residual deviance:  9220.5  on 7660  degrees of freedom
AIC: 9228.5

Number of Fisher Scoring iterations: 4
```

Confusion Matrix and Statistics for our Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2177   73
         1 1585  165
                                          
               Accuracy : 0.5855          
                 95% CI : (0.5701, 0.6008)
    No Information Rate : 0.9405          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.0684          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 0.57868         
            Specificity : 0.69328         
         Pos Pred Value : 0.96756         
         Neg Pred Value : 0.09429         
             Prevalence : 0.94050         
         Detection Rate : 0.54425         
   Detection Prevalence : 0.56250         
      Balanced Accuracy : 0.63598         
                                          
       'Positive' Class : 0               
```

R statistical programming code.

```
Warning in readLines(conn): incomplete final line found on 'CoIL.r'
```

```
# 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)
  }
}
```

```r
## 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 - 49",
  "2" = "f 50 - 99",
  "3" = "f 100 - 199",
  "4" = "f 200 - 499",
```

```
    "5" = "f 500 - 999",
    "6" = "f 1000 - 4999",
    "7" = "f 5000 - 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(LIKELY_CUSTOMERS = ifelse(MOSTYPE %in% set_to_1, 1, 0)) %>%
    mutate(LIKELY_CUSTOMERS = as.factor(LIKELY_CUSTOMERS)) %>%
    mutate(DRIVEN_GROWERS = ifelse(MOSHOOFD == "2", 1, 0)) %>%
    mutate(DRIVEN_GROWERS = as.factor(DRIVEN_GROWERS)) %>%
    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 likely customer types
MOSTYPE_crosstab <- up_train %>%
  select(CARAVAN, MOSTYPE) %>%
  table() %>%
  data.frame()

MOSTYPE_crosstab <- MOSTYPE_crosstab %>%
  group_by(MOSTYPE) %>%
  summarise(total = sum(Freq)) %>%
  merge(MOSTYPE_crosstab) %>%
  mutate(share = Freq / total) %>%
  filter(CARAVAN == 1, share > 0.5) %>%
  arrange(desc(share)) %>%
  select(MOSTYPE, share)

MOSTYPE_crosstab

## Model Building & 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
# specificity = the specificity of the CARAVAN = 1 predictions

  # Generate the predicted outcome
  if(!predictions){
    glm_predictions <- suppressWarnings(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 specificity of the model's CARAVAN = 1 Predictions
  specificity <- correct / nrow(data[data$CARAVAN == 1,])

  # Return the data as a list
  return(list("cm" = cm, "correct" = correct, "specificity" = specificity))
}

robust_results <- function(model_formula, correction = "upSample", n_tries =
250){
## Trains and evaluates the model multiple times
#
# INPUTS
#
# model_formula = The formula for the logit model
# correction (optional) = Correct for imbalanced data (i.e. upSample, downSam
ple, none)
# n_tries (optional) =  The number of runs (250 default)
#
# RETURNS (data.frame)
#
# seed = random number seed
# correct = the number of correct CARAVAN = 1 predictions
# specificity = the specificity of the CARAVAN = 1 predictions

  # Convert the formula from a string
  model_formula <- as.formula(model_formula)
  # Begin the loop
  for(seed in 1:n_tries){
      set.seed(seed)
      # Because some models fail we need to use a try except
      success = tryCatch({
        # Split the data
        train_index <- createDataPartition(df$CARAVAN, p = .7, list = FALSE)
        train <- df[train_index,]
        test_df <- df[-train_index,]
        if(correction == "upSample"){
          # Correct the data imbalance through over sampling
          training_df <- upSample(x = select(train, -CARAVAN),
                                  y = train$CARAVAN,
                                  yname = "CARAVAN")
        } else if(correction == "downSample"){
          # Correct the data imbalance through under sampling
          training_df <- downSample(x = select(train, -CARAVAN),
                                    y = train$CARAVAN,
                                    yname = "CARAVAN")
        } else {
          # No correction
```

```
          training_df <- train
        }
        # Build the model
        model <- glm(model_formula,
                     family = binomial(link = "logit"),
                     training_df)
        # See how it preforms
        results <- score_model(model, test_df)
        # Store the results
        temp <- data.frame("seed" = seed,
                           "correct" = results$correct,
                           "specificity" = results$specificity)
        if(exists("the_results")){
          the_results <- bind_rows(the_results, temp)
        } else {
          the_results <- temp
        }
      }, error = function(e) {
        # Something bad happened
        return(FALSE)
      })
  }
  # Return the data.frame of results
  return(the_results)
}

### Model 1 - Top 5 Important Variables from Random Forest
model1 <- glm(CARAVAN ~ MOSTYPE + PPERSAUT + MOSHOOFD + PBRAND + APERSAUT,
              family = binomial(link = "logit"),
              up_train)
model1_results <- score_model(model1, test)
model1_results$specificity
model1_robust_results <- robust_results("CARAVAN ~ MOSTYPE + PPERSAUT + MOSHO
OFD + PBRAND + APERSAUT")
summary(model1_robust_results$specificity)

### Model 2 - Likely Customers and Car Policies Contribution Level
model2 <- glm(CARAVAN ~ LIKELY_CUSTOMERS + PPERSAUT,
              family = binomial(link = "logit"),
              up_train)
model2_results <- score_model(model2, test)
model2_results$specificity
model2_robust_results <- robust_results("CARAVAN ~ LIKELY_CUSTOMERS + PPERSAU
T")
summary(model2_robust_results$specificity)

### Model 3 - Likely Customers and Car Policies Contribution Level and whethe
r or not they are a driven grower
model3 <- glm(CARAVAN ~ LIKELY_CUSTOMERS + PPERSAUT + DRIVEN_GROWERS,
```

```
              family = binomial(link = "logit"),
              up_train)
model3_results <- score_model(model3, test)
model3_results$specificity
model3_robust_results <- robust_results("CARAVAN ~ LIKELY_CUSTOMERS + PPERSAU
T + DRIVEN_GROWERS")
summary(model3_robust_results$specificity)

## Final Model Accuracy
final_model <- score_model(model3, eval)
final_model$correct
final_model$specificity

## Test Final Model
set.seed(42)
down_train <- downSample(x = select(train, -CARAVAN),
                         y = train$CARAVAN,
                         yname = "CARAVAN")
model3_down <- glm(CARAVAN ~ LIKELY_CUSTOMERS + PPERSAUT + DRIVEN_GROWERS,
                   family = binomial(link = "logit"),
                   down_train)
model3_down_robust_results <- robust_results("CARAVAN ~ LIKELY_CUSTOMERS + PP
ERSAUT + DRIVEN_GROWERS", "downSample")

model3_down_score <- score_model(model3_down, eval)
```

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.