Final Project

Predictive Data Analytic Model

Michael Surdek

Southern New Hampshire University

DAT 640: Predictive Analytics

Professor Tom Woolman

March 6, 2022

# Table of contents

3……Organizational Background

3……Analytic Question and Plan

4……Specifications

5……Predictive Algorithm Recommendation

7……Data Analytic Tools

7……Data Set

8……Gradient Boosting Machine

11……Model Evaluation

14……Scoring Engine

# Organizational Background

            The Insurance Company (TIC) hopes to apply computational intelligence and statistical learning techniques in order to better understand its customers and thereby improve its marketing efforts. TIC provides many types of insurance including automobile, life, property, and social security. The organization has a large customer base, and in the past has seen less than stellar results from trying to promote all of its products to each and every customer, regardless of whether they already hold that type of policy.

            The Insurance Company could implement a better strategy of targeting its marketing campaigns. Promotional emails are currently sent to everyone on their list, including active customers. Spamming these individuals can make them more likely to cancel their insurance policies and less likely to purchase additional insurance. With information from a statistical model based on their customer data, TIC would be able to avoid its active customers and waste fewer resources. In this initiative, they will address this problem with a focus on caravan, or mobile home, insurance.

            The Insurance Company can implement the model into their process by deploying a scoring engine on new data. The results of this initial analysis can be measured in terms of marketing conversion rates and insurance sales figures. If it turns out to improve their caravan insurance business, then a similar strategy could be utilized for the rest of the insurance types, which would add significant value and help the entire organization.

# Analytic Question and Plan

            The question that TIC plans to answer is how to identify which customers are likely to already hold a certain policy. With the demographic, economic, and insurance coverage data that TIC has on hand, they will be able to make predictions about their prospects current coverage status. “The underlying problem is to find the subset of customers with a probability of having a caravan insurance policy above some boundary probability. The known policyholders can then be removed and the rest receives a mailing.” (van der Putten, 2000) The predictions that the model puts out will be a simple yes or no.

            To answer this question, TIC will develop a binary classification model that predicts whether or not a specific customer has a caravan insurance policy One of the benefits of applying computational intelligence to these tasks is that multiple models can be generated automatically and they can be compared to each other. The model that is best for each task will be the one with the highest quality, meaning “hit rate, which should be maximized, and complexity, which should be minimized.” (Kim, 2000) Another benefit of this approach is that the results of the model can be directly integrated into the company’s marketing software and automatically enhance their customer relationship management efforts for an extended period of time.

# Specifications

            There are many binary classification algorithms that the Insurance Company could choose, including “logistic regression, decision trees, support vector machines, Naive Bayes’ classifiers, neural networks, and…k-nearest neighbor algorithm.” (TechVidvan, 2022) These are all forms of predictive analytics, or supervised learning techniques, meaning that “the historic data from which we build our models will already have associated with it speciﬁc outcomes.” (Williams, 2011) Each of these algorithms serves a specific purpose, so selecting the proper technique is a matter of understanding the needs and nuances of the research problem and then identifying the algorithm that can accomplish the task with minimal added complexity.

            A logistic regression model is one approach that would quantify the likelihood of a customer having caravan insurance, from 0 to 100%. For a binary yes or no output, The Insurance Company could then set a threshold value, above which a customer is deemed to be a caravan insurance policy holder. Therefore, a logistic regression model could suit The Insurance Company’s problem of identifying a subset of customers to remove from the mailing list. However, the limitation of this approach is due to the arbitrary nature of setting a threshold value. For this reason, the proportion that is predicted to be current policy holders will not necessarily be equal to the actual proportion of the population.

            On the other hand, a tree-based model, can be expected to result in a distribution of predictions that is related to the proportion of caravan insurance holders in the training dataset. (Amunategui, 2014) This makes the most sense for The Insurance Company because they want to remove the proper amount of prospects from their mailing list. Removing too few prospects would still lead to spamming, which they are trying to eliminate, and removing too many prospects would lead to missing some potential customers, which is detrimental to the business.

            Another benefit of the decision tree approach is that it will control for relationships and interactions between independent variables. This means that if two or more independent variables are correlated within the data set, only the ones that have the most influence on the dependent variable on their own will be incorporated into the model. (Amunategui, 2014) For example, the second half of the data set includes attributes regarding the current insurance policies held by the customer. This can be either the number of policies held and the monetary contribution for each type of insurance, which are of course directly related to each other. In the binary classification model, the algorithm will determine whether the simple count or the monetary contribution has more influence on the output, and the other will likely be left out of the model unless it also adds valuable information.

# Predictive Algorithm Recommendation

            Rather than a single decision tree, The Insurance Company should algorithmically develop an ensemble of trees whose knowledge can be combined to more accurately identify prospects that already have caravan insurance. Both random forests and boosting are ensemble learning approaches that can be applied to basic modelling techniques including regression and decision trees. A simple model, known as a weak learner, can quickly become a strong model through ensemble learning.

            A random forest builds a certain number of independent models by randomly sampling observations and variables to be considered. The prediction that comes from the majority of the models, or the numerical average, is the prediction of the random forest. Boosting also builds multiple models, but they are not independent. “The basic distinguishing characteristic of the boosting approach…is that after building one model any observations that are incorrectly classiﬁed by that model are boosted.” (Williams, 2011) The boosting process begins with equal weighting across observations, and the weights are adjusted based on the whether or not the prediction from each successive model is correct. “Subsequent trees help us to classify observations that are not well classified by the previous trees. Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models.” (Singh, 2018) In a boosting approach, not only are the observations weighted within the trees, but the trees are weighted in the final model based on their individual accuracy.

            The caret package in R provides the gbm method for training a gradient boosting machine model. Caret also provides many arguments and controls that can be used to tune the parameters and refine the model. Sampling preferences, such as cross validation, are passed into the train control. Tree building options, such as the number of trees, the maximum depth of the tree, and the minimum number of observations of an eligible node, are passed into the tune grid.

# Data Analytic Tools

            There are many tools that are capable of facilitating implementation of a predictive algorithm such as a gradient boosting machine. These tools vary in accessibility, performance, and customization. Simple models can be developed in point-and-click tools such as Excel and Minitab. For additional options, improved capabilities, or streamlined and reproducible analyses, code-based tools such as Python’s scikitlearn module or R’s caret package are the best choice. Working with programming languages is best not only for building the model, but also importing and transformating the data beforehand. R even provides the Rattle package, which is a graphical user interface that logs the code used to build the model. Additionally, scoring engines can be deployed directly through Python and R, optimizing the implementation of the analysis for business purposes.

# Data Set

            The data that The Insurance Company can utilize was supplied by Sentient Machine Research (SMR) group based out of Amsterdam, the Netherlands. The data is stored across three files. The first file contains training data, which consists of 5822 records of 86 attributes, including the target variable “CARAVAN:Number of mobile home policies.” The second file contains validation data, another 4000 records of 85 attributes. The targets of the validation data are held out in the third data file.

            There are two types of attributes in the datasets. Half of the attributes contain economic or demographic information, such as religion, marriage status, and income. The other half of the attributes contain product ownership information, which includes the number of policies as well as the monetary contributions towards each type of insurance policy: automobile, life, boat, etc. (van der Putten, 2000)

            Each attribute in the data set is recorded as an integer value. Most attributes range from 0 or 1 to another single digit integer such as 5, 8, or 9. The first two attributes are the exceptions, ranging up to 41 and 10, respectively.

# Gradient Boosting Machine

            The first step towards building a gradient boosting machine in R is loading the required packages and the insurance data set. The data file is read from my github profile for reproducibiliy purposes. The only transformations are simply assigning the 86 column names and converting the target variable, CARAVAN, to a factor with 2 levels, which is necessary for a binary classification model.

library(dplyr)  
library(readr)  
library(rpart)  
library(caret)  
library(ada)  
library(mlbench)  
library(MLmetrics)  
library(gbm)  
library(curl)  
library(MLeval)

coil <- read.delim(curl('https://raw.githubusercontent.com/msurdek/School/main/DAT%20640%20Predictive%20Analytics/Assignments/Project/Milestone%204/ticdata2000.txt'), header=FALSE)  
  
colnames(coil) <- 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")  
  
coil <- coil |> mutate(CARAVAN = as.factor(CARAVAN))

            The data set is now prepared to be fed into the training function for the gbm model. First, the control and tuning options are stored as variables that will be passed to the training function. The control will use 5-fold cross validation. The algorithm will generate 1000 trees with a maximum depth of 30 nodes and a minimum number of observations in a node of only 1. The sampling seed is also set so that the same samples will be generated every time. Additionally, within the training function, the gbm method is specified and the distribution of the target variable is bernoulli, which is used for logistic regression of a binary outcome.

fitControl <- trainControl(method="cv",  
 number=5,  
 returnResamp = "all")  
  
tuneControl <- data.frame(.n.trees=1000,  
 .shrinkage=0.01,  
 .interaction.depth=30,  
 .n.minobsinnode=1)

set.seed(10)  
model2 <- train(CARAVAN~.,  
 data=coil,  
 method="gbm",  
 distribution="bernoulli",  
 trControl=fitControl,  
 verbose=F,  
 tuneGrid=tuneControl)

model2

## Stochastic Gradient Boosting   
##   
## 5822 samples  
## 85 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 4657, 4658, 4657, 4658, 4658   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.9285468 0.07182122  
##   
## Tuning parameter 'n.trees' was held constant at a value of 1000  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.01  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 1

            The model is now encoded as an object and some further information can be pulled from its elements. A summary of the model shows the variables with the largest relative influence on the target variable, CARAVAN. There are 6 variables with a relative influence greater than 2.5. These are, in descending order, customer subtype (MOSTYPE), car policy contribution (PPERSAUT), fire policy contribution (PBRAND), third party insurance contribution (PWAPART), protestant (MGODPR), and no religion (MGODGE).

head(summary(model2),12)

| *var* | *rel.inf* |
| --- | --- |
| *MOSTYPE* | *6.676186* |
| *PPERSAUT* | *6.324712* |
| *PBRAND* | *5.967528* |
| *PWAPART* | *4.447999* |
| *MGODPR* | *2.639266* |
| *MGODGE* | *2.637284* |
| *MKOOPKLA* | *2.190468* |
| *MBERMIDD* | *2.093528* |
| *MBERARBG* | *2.043321* |
| *MOPLHOOG* | *2.006023* |
| *MINK3045* | *1.980225* |
| *MOPLMIDD* | *1.947163* |

# Model Evaluation

            There are numerous metrics available to measure the effectiveness of a predictive model, including accuracy, sensitivity, specificity, recall, precision, log loss, and area under curve. These “will allow us to understand what to expect when we use the model to score new observations.” (Williams, 2021) Evaluation can help to decide between competing models and to improve an individual model.

            The confusion matrix of the gradient boosting machine model “shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data.” (Sayad, 2017) The simple counts from the confusion matrix can be used to calculate the accuracy, sensitivity, specificity, positive predictive value, and negative predictive value. For example, the specificity is the proportion of true positives out of all positive predictions. For this model, it is 192 / 156 + 192, or 55%.

pred <- predict(model2, coil, na.action = na.pass)  
  
confusionMatrix(pred, coil$CARAVAN)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 5469 156  
## 1 5 192  
##   
## Accuracy : 0.9723   
## 95% CI : (0.9678, 0.9764)  
## No Information Rate : 0.9402   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6912   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9991   
## Specificity : 0.5517   
## Pos Pred Value : 0.9723   
## Neg Pred Value : 0.9746   
## Prevalence : 0.9402   
## Detection Rate : 0.9394   
## Detection Prevalence : 0.9662   
## Balanced Accuracy : 0.7754   
##   
## 'Positive' Class : 0   
##

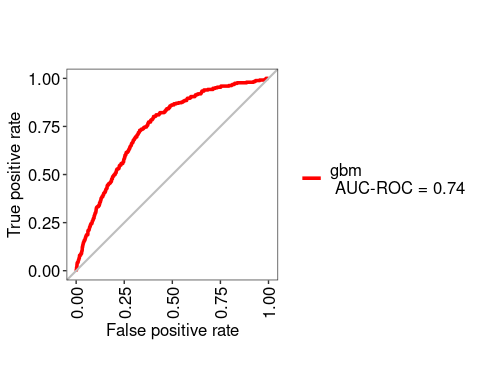
            Additional evaluation metrics can be generated from the multiClassSummary function of the caret package. For example, the log loss, which accounts for the confidence of each prediction, is 0.075, and the Kappa statistic, which “represents the extent to which the data collected in the study are correct representations of the variables measured,” (McHugh, 2012) is 0.691.

results <- predict(model2, coil, na.action = na.pass, type = "prob")  
  
results$obs <- coil$CARAVAN  
  
results$pred <- predict(model2, coil, na.action = na.pass)  
  
multiClassSummary(results, lev = levels(results$obs))

## logLoss AUC prAUC Accuracy   
## 0.07459029 0.99414867 0.96168582 0.97234627   
## Kappa F1 Sensitivity Specificity   
## 0.69124517 0.98549419 0.99908659 0.55172414   
## Pos\_Pred\_Value Neg\_Pred\_Value Precision Recall   
## 0.97226667 0.97461929 0.97226667 0.99908659   
## Detection\_Rate Balanced\_Accuracy   
## 0.93936791 0.77540536

            Finally, another key evaluation metric is the area under the curve, which is derived from an ROC chart that plots the true and false positive rate. The AUC of the gradient boosting machine model is 0.74. AUC serves as useful measure to compare the performance of multiple models. The curves can be included and compared on the same ROC chart. This may show that one model is best across the board, or even that one model performs better at a lower or higher false positive rate than others, which may be preferable based on the objective of the model.

auc <- evalm(model3, gnames = "gbm", plots = "r")



# Scoring Engine

            For The Insurance Company to make use of this predictive model, they will have to implement a scoring engine that applies the model to new data and uses the predictions to guide their strategy. In this situation, TIC is hoping to email its prospects with a caravan insurance promotion, but to avoid those who already hold that policy. A scoring engine could be implemented by feeding information about potential prospects, which must match the contents of the training data, into the gbm model. The results of the scoring engine would be a yes or no output that predicts caravan insurance ownership. The prospects that the model predicts “yes” could then be removed from the promotional mailing list.

            The expected results of the scoring engine can be estimated from some of the evaluation metrics above. For example, based on the confusion matrix, of the 5473 observations where the customer did not already hold caravan insurance, only 5 were not predicted as such. On the other hand, 192 of the 348 observations where the customer did already hold caravan insurance were predicted correctly. The model accurately predicts 99.9% of non-caravan insurance owners and 55.2% of caravan insurance owners, for an overall accuracy rate of 97.2%. If The Insurance Company used this model to advise their marketing strategy, they could then expect to successfully eliminate 55% of current caravan policy holders, at the cost of missing less than one in a thousand potential prospects.

References

Amunategui, Manuel. 2014. Modeling 101 - Predicting Binary Outcomes with R, gbm, glmnet, and {caret}. Github. Retrieved from https://amunategui.github.io/binary-outcome- modeling/#

Kim, YongSeog and Street, Nick. *CoIL Challenge 2000: Choosing and Explaining Likely Caravan Insurance Customers*. University of Iowa Management Sciences Department. May 26, 2000. Retrieved from https://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/STREET~1.pdf

McHugh, Mary. 2012. *Interrater reliability: the kappa statistic.* Biochem Med (Zagreb). Retrieved from https://pubmed.ncbi.nlm.nih.gov/23092060/

Sayad, Saed. 2017. Model Evaluation – Classification. Saedsayad.com. Retrieved from

https://web.archive.org/web/20170107084843/http://saedsayad.com/model\_

evaluation\_c.htm

Singh, . 2018. *Understanding Gradient Boosting Machines*. Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-gradient-boosting-machines- 9be756fe76ab

TechVidvan. 2022. R Classification – Algorithms, Applications and Examples. TechVidvan. Retrieved from https://techvidvan.com/tutorials/classification-in-r/

van der Putten, Peter. de Ruiter, Michel. van Someren, Maarten. 2000. *CoIL Challenge 2000 Tasks and Results: Predicting and Explaining Caravan Policy Ownership*. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report. Retrieved from https://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/PUTTEN~1.pdf

Williams, Graham. 2011. Data Mining with Rattle and R : The Art of Excavating Data for Knowledge Discovery. Springer New York. Retrieved from http://ebookcentral.proquest.com/lib/snhu- ebooks/detail.action?docID=770229