Insurance Cross Selling (Kaggle)

Ellis Hodgdon

Nov 18 2024

Problem Definition

As a result of the acquisition of a smaller company by this company, management has decided to investigate cross-selling of their insurance products and rewriting the business plan as necessary. To accomplish this, a data analysis is ordered to determine if cross-selling would be beneficial to the combined companies. The goal of this analysis is to predict which customers respond positively to an automobile insurance offer. A dataset has been developed that contains information about cross-selling in the insurance industry.

Input the training dataset and data prep

Kaggle provides two datasets – training and test – but the test dataset is only used to judge the developed algorithm in a competition and will be ignored here since it is missing the *Response* column. We will work with the entire training dataset of 907,971 rows and then split it into two (or more) datasets for the analysis The training dataset will be 90%, the testing set will be 10% of the original dataset. The division of the training dataset and the testing dataset is somewhat arbitrary. The ratios of 70/30, 80/20, and 90/10 were tried, with the 90/10 generating the best accuracy.

Method

These two different datasets will be used this analysis. The training set *(training_set)* will be used to develop the algorithm, and the testing of this algorithm will be done by using the testing_set (*testing_set*). The testing dataset is constant across all models and it should be noted that no rows are duplicated across any datasets. A feature of this analysis program is that the number of test datasets can be changed with averaging of the results. For this analysis, only one test set was used, but if additional test sets are desired, the number can be changed in the YAML header.

Description of dataset columns

Table 1: Column Names in Dataset

	Name	Description	Class
Identification	id	id	integer
Gender	Gender	Gender	character
Age (in years)	Age	Age	integer
Driver's License Nbr	Driving_License	Driving_License	integer
Region Code	Region_Code	Region_Code	numeric
Previously Insured	Previously_Insured	Previously_Insured	integer
Insured Vehicles Age	Vehicle_Age	Vehicle's_age	character
Vehicle Damage	Vehicle_Damage	Vehicle_damage	character
Annual Premium	Annual_Premium	Annual Premium	numeric
Policy Sales Channel	Policy_Sales_Channel	Policy_Sales_Channel	numeric
Vintage	Vintage	Vintage	integer

	Name	Description	Class
Response	Response	Success	integer

The column *Result* as integer (0 or 1) indicates whether a resale was successful with this customer.

Data Wrangling

Data preparation and cleaning are done by a function so that it can be reused for different test and training datasets, if necessary.

- The function provides the following operations
  - removal of any NAs
  - removal of any rows that contain blanks
  - checks near zero variance and removes any columns that have a near zero variance
  - converts the Gender column to numeric 1 or 0
  - converts the Vehicle_Damage column to numeric
  - converts the vehicle_Age column to a numerical value of the unique values in the column
  - removal of the id column

No NAs were detected.
No blanks were detected.

The near-zero variance test revealed 2 columns that fell into this category. These columns were:

Table 2: Columns Removed

Driving_License
Annual_Premium

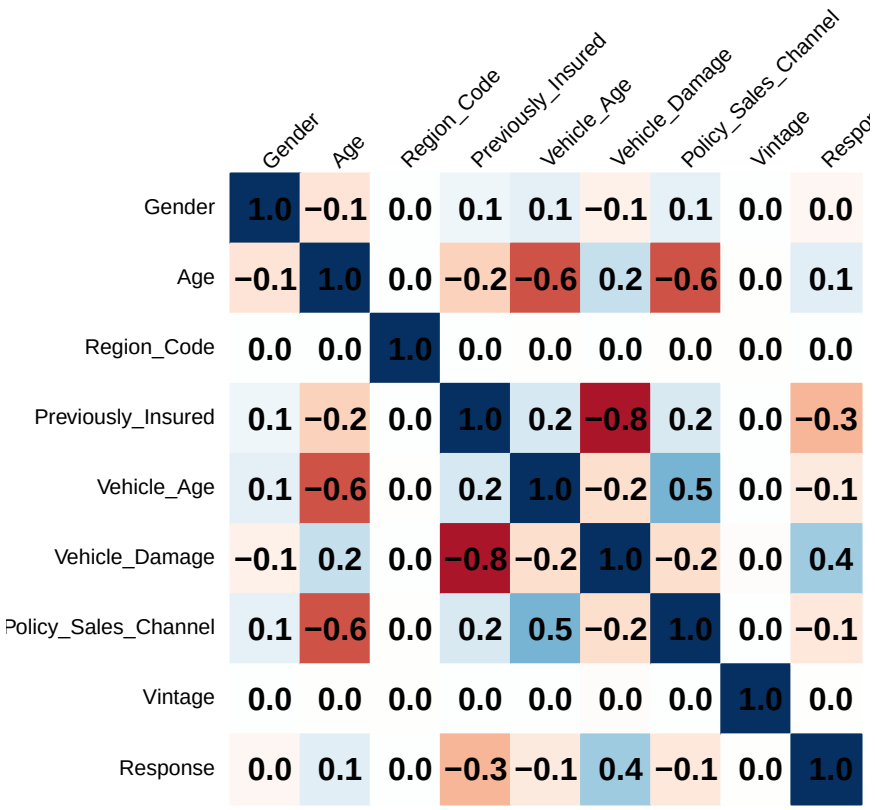
and were removed from the dataset since these columns would not be good features for the training and testing datasets. The objective of this preprocessing to to make the dataset data tidy, which is essential for quality data analyses.

- There are three interrelated rules that make any dataset tidy (Wickham, Centinkaya-Rundel, and Grolemund 2023):
  - ∗ Each variable is a column; each column is a variable
  - ∗ Each observation is a row; each row is an observation
  - ∗ Each value is a cell; each cell is a value

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is not a formal process with a strict set of rules but rather a state of mind where the analyst should investigate every idea that pops up. Some will work out; some will not, but it is an important part of any data analysis because the quality of the data always needs to be evaluated. (Wickham, Centinkaya-Rundel, and Grolemund 2023).

Since much of the data appears to be categorical, to identify if there are any odd correlations between the features, a Spearman test (Rovetta 2020),will be developed. (Another test, the Pearson test is commonly used, but it is generally for continuous data).



There are a few somewhat strong correlations here like Vehicle_Damage and Previously_Insured, but really nothing to upset the decision as to whether to engage in cross product selling or not.

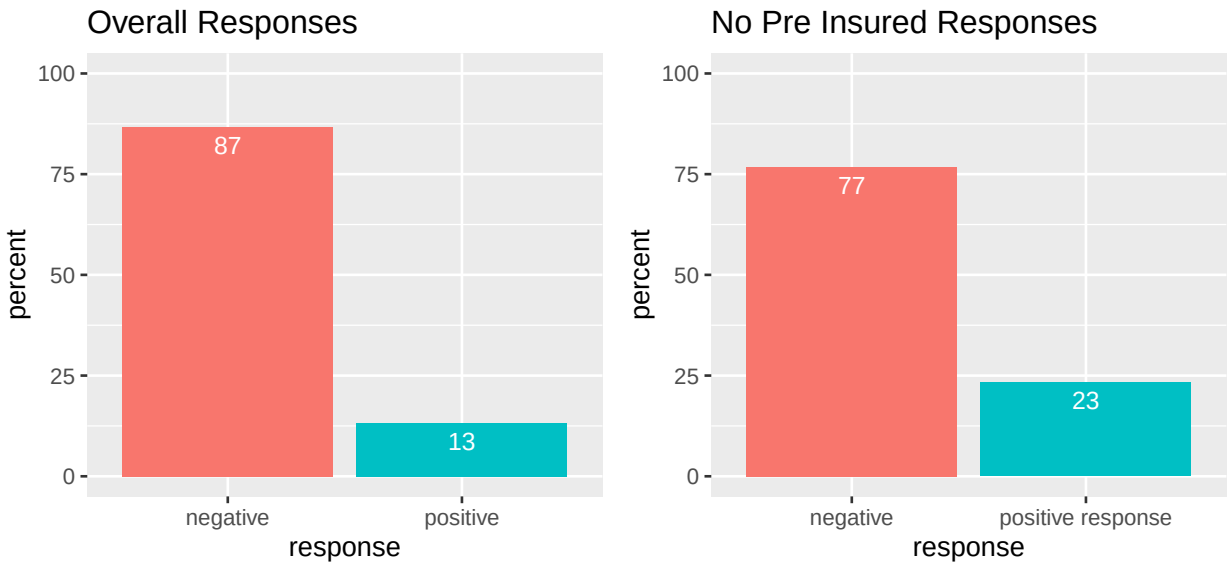
One might suspect that there would be no positive responses when there was no previous insurance applied. To determine if this is true we look at the number of positive responses when the column *Previously_Insured* is 0.

```
num <- sum(training_set$Previously_Insured == 0 & training_set$Response == TRUE)
num
```

```
## [1] 119961
```

The 119,961 positive hits shows that our suspicion is not valid and we need to consider *Previously_Insured* as a feature.

We now consider the *Response* column. What percentage of the responses are positive, that is, how many cross-selling attempts were successful. We then ask the question if there are any vehicles that are not insured that have a positive response.



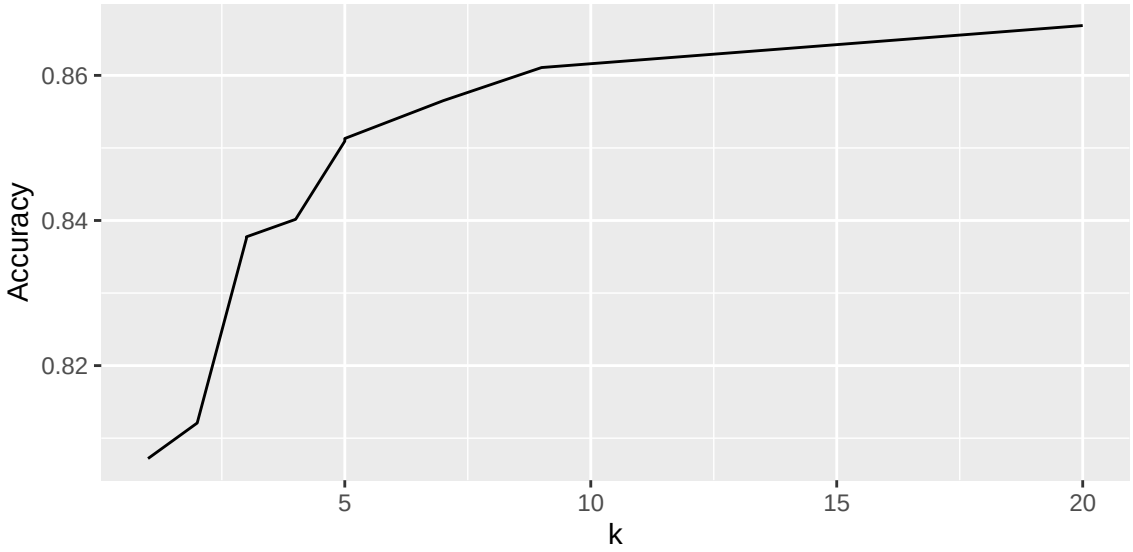
The graph shows that there is a definite component of positive responses from customers that have not purchased insurance before.

Modeling

We will proceed with the analysis using several different models and in the end, choose the most accurate.

k-NN (k-Nearest Neighbor)

We use the k-NN method of training which has a parameter of k. We examine the accuracy of the training for various values of k. We initially try a k in the set of 3, 5, 7, 9, 11



From this second set of iterations, the maximum k value that we should use is 10. The final algorithm is calculated with this value.

Now consider the test sets that we carved from the original data set. How do they match with the algorithm (function *predict*)? Here are the accuracy results for the k-NN model:

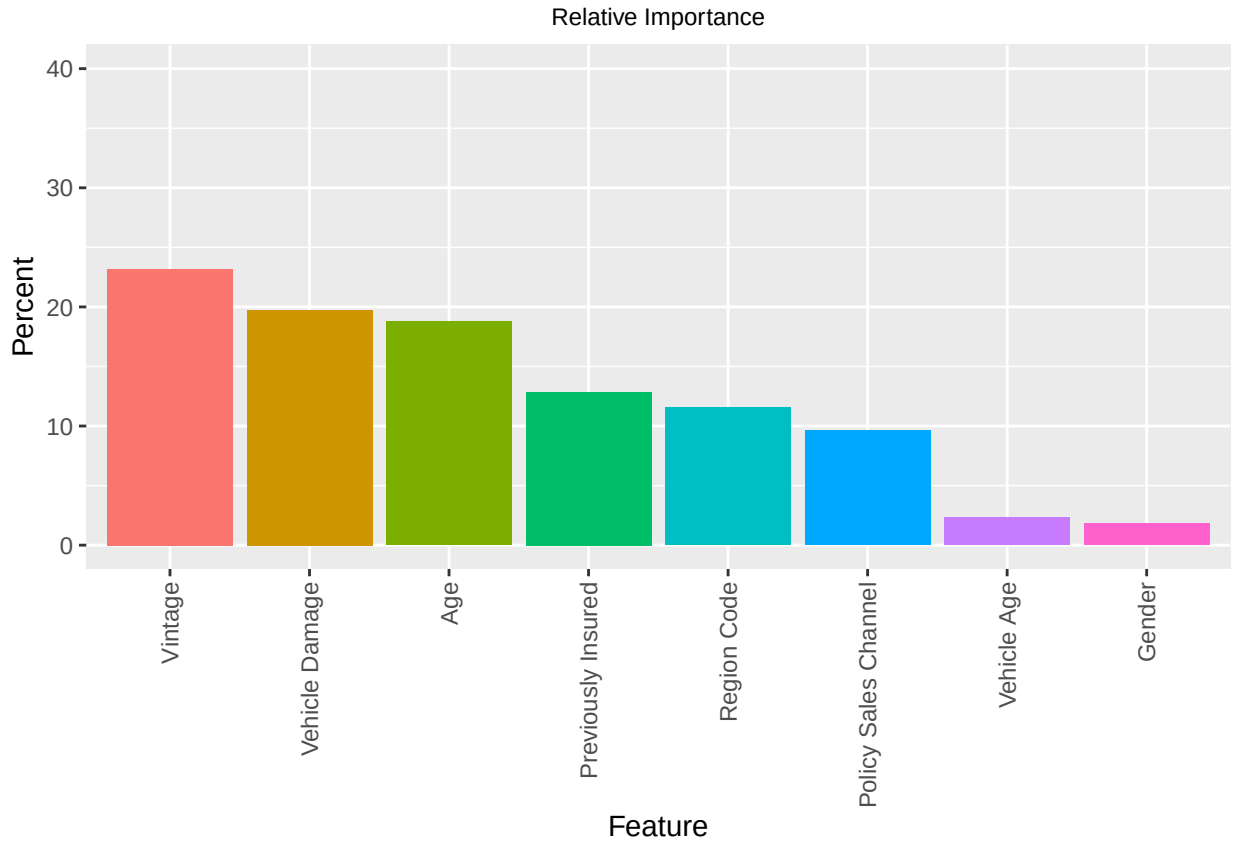
Accuracy	Sensitivity	Specificity
0.8613	0.9657	0.1113
Average	0.8613	

This algorithm was based on the k-NN (nearest neighbor). Other models will be tested for comparisons. The first alternative to attempt is random forest which is a compute-intensive model. There are several parameters that can be tuned to adjust the model and we will need to build a multitude of trees. As a result of this, we will only use a five-fold cross validation. The results of the 1 test cases are as follows:

Random Forest

Accuracy	Sensitivity	Specificity
0.8775	0.9967	0.0214
Average	0.8775	

The average of these 1 test cases is 0.8775033. One feature of the random forest training is the development of the importance of the various features.



We can see from this graph what are the most important features in this random forest analysis. ### Ensemble

When we combine the two models that we have tried (knn and random forest) to determine if the combination is better than either one and run the 1 test sets. After going through a similar process as was done for k-NN

and random forest, produces the follow table of the tests. However, the ensemble accuracy turns out to be 0.875573673597107 which is below the random forest accuracy of 0.8775033.

Accuracy	Sensitivity	Specificity
0.8756	0.9939	0.0257
Average	0.8756	

Naive Bayes

Another model that was tried was the *naive Bayes* model which seeks the model the distribution of inputs but does not learn which features are most important. Again, using the standard train set, we find that the accuracy from the confusion matrix is , which is still lower that was obtained from either the *k-NN* model or the *random forest* model.

Accuracy	Sensitivity	Specificity
0.7113	0.6854	0.8977
Average	0.7113	

General Linear Model (glm) The glm has several families available for the model: binomial, gaussian, quasi, and quasibinomial, and others. An analysis was performed on each of these families and the results were:

Model	Accuracy	Sensitivity	Specificity	MinMax
glm (binomial)	0.6296	0.8262	2e-04	Maximum
glm (gaussian)	0.6296	0.8262	2e-04	
glm (quasi)	0.6296	0.8262	2e-04	
glm (quasibinomial)	0.6296	0.8262	2e-04	
Model	Average	0.6296		

Conclusion

There are over 21,000 models ("Contributed Packages," n.d.) that are available from CRAN to determine the best fit for this data. The method recommended for binary data is the general linear regression model. The results from using this model were disappointing.

A summary of the accuracy results for the models the were tested is:

Model	Accuracy	Sensitivity	Specificity
rf	0.8775	0.9967	0.0214
ensemble	0.8756	0.9939	0.0257
knn	0.8613	0.9657	0.1113
naiveBayes	0.7113	0.6854	0.8977
glm (binomial)	0.6296	0.8262	0.0002
glm (gaussian)	0.6296	0.8262	0.0002
glm (quasi)	0.6296	0.8262	0.0002
glm (quasibinomial)	0.6296	0.8262	0.0002

Comparing the average for the different models, we determine that the rf model gives the best accuracy at 0.8775033. At this accuracy, the organization is encouraged to begin cross-selling acknowledging that there is some risk involved.

- If management decides to engage in cross-selling, the Spearman Correlation Map offers a couple of suggestions:
  - ∗ Approach customers that have not had a vehicle damaged
  - ∗ Avoid those that have been previously insured
  - ∗ Age is a small factor, but concentrate on the older customers first.

This analysis should be completed periodically to ensure accurate results.

Appendix

System Information RStudio: version 2024.09.0 Build 379
R: Version 4.3.2 (2023-10-21 ucrt) – "Eye Noise"
Windows 11 Pro Version 22H2 Build 22621.4317 64 bit operating system 64 bit processor
Dell 5431 Processor 2.20 GHz Memory 32 GB

Configuration for this analysis

Parameter	Value
number_of_test_sets	1
testing_set_percentage	testing_set_percentage
dataset_percentage	10

```
## [1] "Elapsed time for this analysis: 261.99 mins"
```

```
## [1] "Finished at: 15:23 PM"
```

Bibliography

"Contributed Packages." n.d. https://cran.r-project.org/web/packages/.

Rovetta, A. 2020. "Raiders of the Lost Correlation: A Guide on Using Pearson and Spearman Coefficients to Detect Hidden Correlations in Medical Sciences." *Cureus*. https://doi.org/10.7759/cureus.11794.

Wickham, H., M. Centinkaya-Rundel, and G. Grolemund. 2023. "R for Data Science." Sebastopol, CA: O'Reilly Media, Inc.