Ray Karpman
May 10, 2022

Springboard Capstone 3: Customer Segmentation for Caravan Insurance

1. Background and Problem Statement

The goal of this project is to help a hypothetical insurance company develop a cost-effective marketing strategy for caravan (mobile home) insurance. Caravan insurance is a relatively niche product. Most consumers will never own a caravan, so marketing to them would be a waste of money. Our client wants to identify groups of customers that are likely to purchase caravan insurance, so that they can design advertising campaigns that will reach those groups and appeal to their sensibilities.

To help the client design a targeted marketing approach, I decided to try customer segmentation using the k-means clustering algorithm. My strategy is to group the client's customer data into natural clusters using k-means, compare rates of caravan ownership between two clusters, and investigate the characteristics that distinguish each cluster from the others. An optimal clustering should reflect natural patterns in the data, and contain clusters with meaningfully different rates of caravan insurance ownership. In addition, for the clustering to provide actionable insights, it should be interpretable. That is, one should be able to give a concise and informative description of the types of customers that fall into each cluster.

2. Data Understanding and Cleaning

The data set used in this project was originally created by Dutch company Sentient Machine Research, based on real business data (van der Putten and van Someren, 2000). This data set is now part of the UCI Machine Learning Repository and has been posted, together with a detailed codebook, on kaggle.com (UCI 2017).

The original data set had 9822 rows and 87 columns. One column recorded whether the row was part of the test or training set of a previous competition. This was not relevant to the project, so I dropped it. Feature names were abbreviations based on Dutch phrases. Fortunately, a detailed set of keys was posted on kaggle.com.

Each record in this data set corresponded not to a single customer or household, but to a Dutch postal code. The target variable, CARAVAN, was 0 for postal codes with no caravan insurance policies, and 1 for postal codes with at least one caravan insurance policy. The data was highly unbalanced, with only 6.77% of entries in the CARAVAN column being 1's.

All features were categorical, coded as integers. There were two nominal features: customer main type (MOSHOOFD) and customer subtype (MOSTYPE). These are pre-existing customer segments, which vary substantially both in number and in rates of Caravan insurance ownership. Customer main type had 10 categories, coded as integers. Subtype had 41 categories, also coded as integers.

Ordinal features included number of houses (1-10), average household size (1-6), and average age, binned by decade. Of the remaining features, 38 corresponded to the percent of a postal code that was in some demographic group. These were binned and coded as integers, with 0 and 9 corresponding respectively to 0% to and 100%. Other values were grouped into roughly equal bins, coded 1-8.

The remaining 41 features correspond to either total contribution to or number of certain insurance policies in a postal code. These were grouped into bins coded 0-9, whose size increased non-linearly.

The data had no values coded as missing. There was only one row with a value not in a code book (which I dropped). I discovered 1442 duplicate rows. While it's possible that these corresponded to postal codes which happened to have identical characteristics, I assumed they were likely included in error and chose to drop them.

Since all data was categorical, there was no issue with outliers in the typical sense. However, there were "categorical outliers," or very rare categories. I viewed a category as rare if it appeared fewer than 85 times in the data set, since I wanted to ensure at least 5 expected entries per cell when performing a χ-squared test for differences in caravan ownership rates. For the nominal variable MOSTYPE (customer subtype), I grouped together adjacent rare categories with similar descriptions.

For ordinal variables, I wanted to strike a balance between grouping together rare categories and preserving the natural ordering of the data. After plotting frequency distributions for all ordinal variables, I noticed that many were either mound-shaped, or skewed with a tail to the left or right. My strategy here was to cap outliers. For each variable, I capped outliers below by finding the lowest value with 85 values total at or below it, and replaced all such values with that minimum. I capped high outliers in a similar way. After completing this process, I realized that 16 features now had only one unique value; these features were extremely low-variance to begin with, so I dropped them. The cleaned data frame had 8379 rows and 70 columns.

3. Exploratory Data Analysis

My first step was to plot a heatmap of the correlations between ordinal variables. Since these variables are categorical, I used Spearman's rank correlation rather than the usual Pearson correlation. From the heatmap, it was clear that there were many pairs of highly-correlated features. Highly correlated or redundant features can make models unnecessarily complex and difficult to interpret, and having two many features presents challenges for distanced-based algorithms such as clustering. So, I decided to eliminate highly-correlated features.

During the EDA stage, I tried dropping highly-correlated features manually, by searching for pairs of highly-correlated features and dropping the one which had smaller absolute correlation with the target variable. During the preprocessing stage, I conducted a similar feature-selection process, but this time using a pre-built Python package.

After dropping highly correlated features, I plotted frequency distributions and rates of caravan insurance ownership by category for each of the remaining ordinal variables. As one might expect, rates of caravan insurance ownership tended to increase with various measures of income, and with increased usage of various insurance policies. All correlations between features and the variable CARAVAN were small in magnitude, probably because most values of CARAVAN were 0. However, many of the correlations were statistically significant. For example, the figure below shows that as the percentage of people with low educational attainment in a postal code goes up, rates of caravan insurance ownership go down. This relationship is statistically significant, $p < 0.00001$.

Similar plots showed differences in Caravan insurance ownership by customer type and customer subtype, which were also statistically significant.

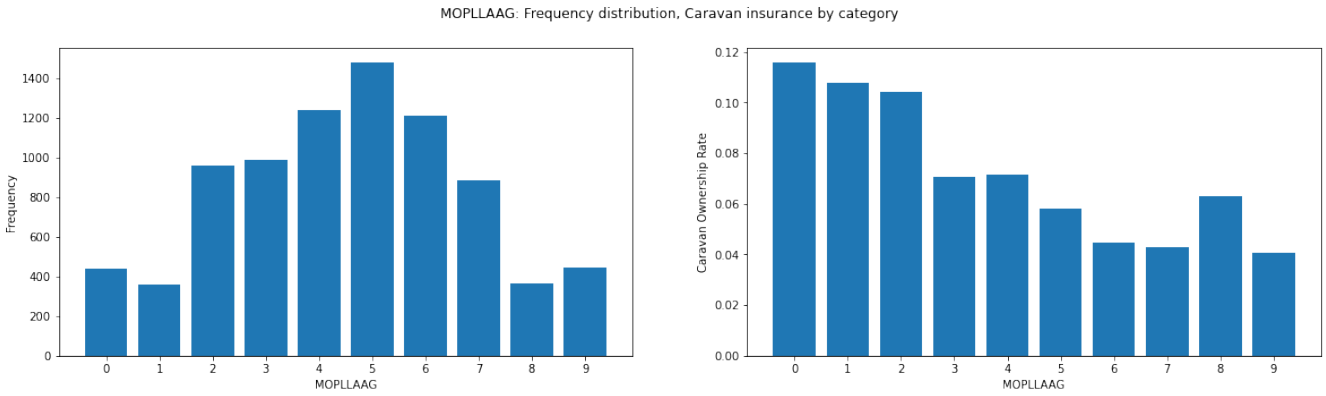

Figure 1: As the percentage of people with low education increases, the proportion of people with Caravan insurance declines.

4. Preprocessing and Feature Selection

All features of the original data set were encoded as integers. For the ordinal features, I kept this encoding. For nominal features, I used numerical encoding, replacing each value of a nominal categorical feature with the rate of caravan insurance ownership for records with that value of the feature. I scaled all features to a mean of 0 and a standard deviation of 1, using sklearn's StandardScaler function.

My next step was feature selection. Highly correlated features can make models unnecessarily complex, and make interpretation difficult. In addition, clustering algorithms may struggle on high-dimensional data. I performed feature selection on the data, using the featurewiz package created by Ram Sheshadri (Sheshadri 2020). This package automatically selects a list of relevant uncorrelated features, using a two step process.

1. Find an uncorrelated list of predictors, using the so-called SULOV algorithm. The algorithm finds pairs of highly-correlated features, and drops the feature in each pair which has a lower mutual information score with the target. Mutual information score is non-parametric, so is applicable to many types of variables.
2. Repeatedly fit simple gradient boosting models using the uncorrelated list of predictors from Step 1. Return a list of variables which had high importance in at least one round of gradient boosting.

With its default options, Featurewiz selected 15 uncorrelated, relevant predictors for use in the modeling step.

5. Modeling

The modeling phase of this project took place in two steps. First, I used the k-means algorithm to cluster the data, and confirmed that my clusters had statistically significant differences in rates of caravan insurance ownership. I then fit decision trees to the feature data, to predict cluster labels assigned in the first step. Finally, I analyzed the decision trees to determine concise descriptions of type of customer found in each cluster. All stages of the modeling process were completed in a Jupyter notebook, using Python's scikit-learn library.

First, I clustered the data using the k-means algorithm. This well-known clustering algorithm treats each record as a point in a many-dimensional space. The algorithm is initialized with k random points, or means. Each data point is then assigned to the nearest of the k points. The centers (means) of the resulting clusters are calculated, and the process repeats, with the calculated cluster means replacing the initial random points. The process of assigning points to clusters and calculating means is then repeated, until the cluster centers converge.

To find an appropriate clustering, I varied both the number of features used in the clustering, and the number of clusters k. When selecting a subset of features to use, I chose the features which had the highest mutual information score (MIS) with the target.

Clustering is an unsupervised learning problem, and there is no hard-and-fast rule for evaluating the correctness of a clustering. I used average silhouette score as my primary metric success. The silhouette score of each point indicates how close that point is to others in its own cluster, compared to points in other clusters. Silhouette score ranges from -1 to 1, with -1 being the worst and 1 being the best.

In addition to average silhouette score, I considered whether a clustering was useful from a business standpoint. For the clustering to convey actionable insights, the number of clusters should be reasonable (I considered between 3 and 7 ideal), and the clusters should be reasonably balanced in size.

I found that using only the top five predictors by MIS gave the highest silhouette scores. Using only two clusters gave the highest silhouette score, while three clusters gave a score that was only slightly lower. Since a clustering with only two clusters may not give enough information to be interesting, I chose five features and three clusters for my final clustering.

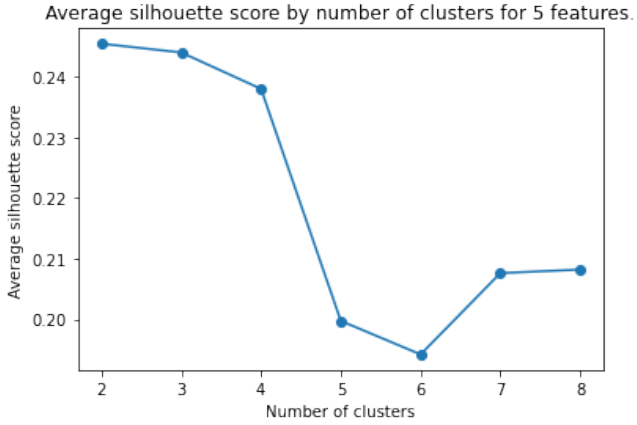

Figure 2: Plotting silhouette score vs. number of clusters

The five features used in the clustering were:

- PPERSAUT: total contribution to car policies
- MOSHOOFD: customer type (10 options)
- MINKGEN: average income
- MOPLLAAG: percent with a low level of education
- MAUT1: percent owning one car
-

I found that the clusters were reasonably balanced in size, and had different rates of caravan insurance ownership, as shown in the table below.

Cluster	Number of zip codes	% with a Caravan policy
Cluster 0	2408	10.71%
Cluster 1	3008	2.76%
Cluster 2	2963	7.66%

A two-tailed χ-squared test for a difference in proportions showed that the differences between the three clusteres were statistically significant.

Having found a reasonable clustering, I used decision trees to help explain the cluster labels. I built a dataframe containing features and cluster labels, without the target column CARAVAN. At first, I tried building a decision tree which would classify each record as belonging to one of the three clusters. Unfortunately, the resulting tree had low accuracy on Cluster 0, the cluster most likely to purchase caravan insurance. To address this problem, I decided to instead fit a separate tree for each cluster. Each tree was responsible for predicting whether a given data point fell into one specific cluster, or not.

During the decision tree phase of the project, my priority was to create trees that were simple and easy to interpret, while maintaining decent predictive performance. I tuned the hyperparameter min_impurity_decrease for each decision tree, using three-fold cross-validation on the training set. Once I had chosen a value of min_impurity_decrease based on the training data, I checked the predictive performance of the resulting decision tree classifier on the test data.

The table below gives hyperparameter values and predictive performance on the test set for the three decision trees. Visualizations of the trees are provided in the next section.

Cluster	min_impurity_decrease	Accuracy	Precision	Recall
Cluster 0	0.01	0.93	0.88	0.88
Cluster 1	0.025	0.93	0.90	0.93
Cluster 2	0.025	0.95	0.91	0.95

6. Analysis and Recommendations

Examining the decision trees which predict each cluster label, one can begin to form a picture of the type of customer that falls into each cluster. Visualizations of the trees themselves are included as an appendix at the end of this report.

Cluster 0 customers are the most likely to purchase caravan insurance. They generally have moderate to high education levels, and come from postal codes with moderate to high average income. They tend to be from the customer types (recorded in the variable MOSHOOFD) more likely to purchase caravan insurance. There are:

- Type 1: Successful hedonists
- Type 2: Driven growers
- Type 3: Average family.
- Type 8: Families with grownups.
- Type 9 Conservative families.

In particular, every customer type which involves the word “family” is associated with a higher rate of Caravan insurance ownership.

Cluster 1 customers are the least likely to purchase caravan insurance. They tend to contribute less to car insurance policies, and come from zip codes with moderate to low average income. The fact that these customers are not buying caravan insurance seems reasonable, as someone who is not spending much on car insurance is unlikely to want to spend money on a niche verhicle insurance product.

Cluster 2 customers own caravan insurance at a rate similar to the population as a whole. They tend to contribute more to car insurance policies, but come from zip codes with lower levels of education. While the demographic profile of this cluster is less clear than for Cluster 0, they may be successful blue-collar workers.

Based on this summary of the three clusters, I would recommend the client focus their efforts on Cluster 0 customers, who are most likely to buy caravan insurance. They might consider advertising in venues that cater to people with higher incomes and higher education levels. Since the family-focused customer types tend to appear in this cluster, the client might consider an advertising campaign showing happy families vacationing together.

The client may also consider a secondary focus on Cluster 2 customers. Since these customers tend to spend more on car insurance, advertising or promotions directed at current car-insurance customers may be successful. They might also focus on advertising venues more likely to appeal to skilled tradesmen and other financially successful blue-collar workers.

7. Further Work

There are some important limitations in this project. The most obvious issue is that so few features were used in the final clustering, which means we have limited information about each cluster. One way to address this might be to examine differences between other features between our three clusters. This might give us greater insight into the characteristics of each group. Another option could be to include more features in the clustering, at the cost of a potentially lower silhouette score. Finally, it may help to rerun the pipeline, but be less aggressive about removing correlated features. While truly redundant features are not helpful, the presence of some correlated features could lead to stronger clusterings.

Another limitation, from a technical standpoint, is that our clustering is relatively weak. The silhouette score for our final clustering was only 0.24, and indication that clustering could be artificial. It may be helpful to explore other clustering algorithms, such as density-based or mean-shift clustering. Different algorithms perform well on different data sets, and any natural groupings in our data might be better captured by a different algorithm.

Appendix: Visualizations of Decision Trees

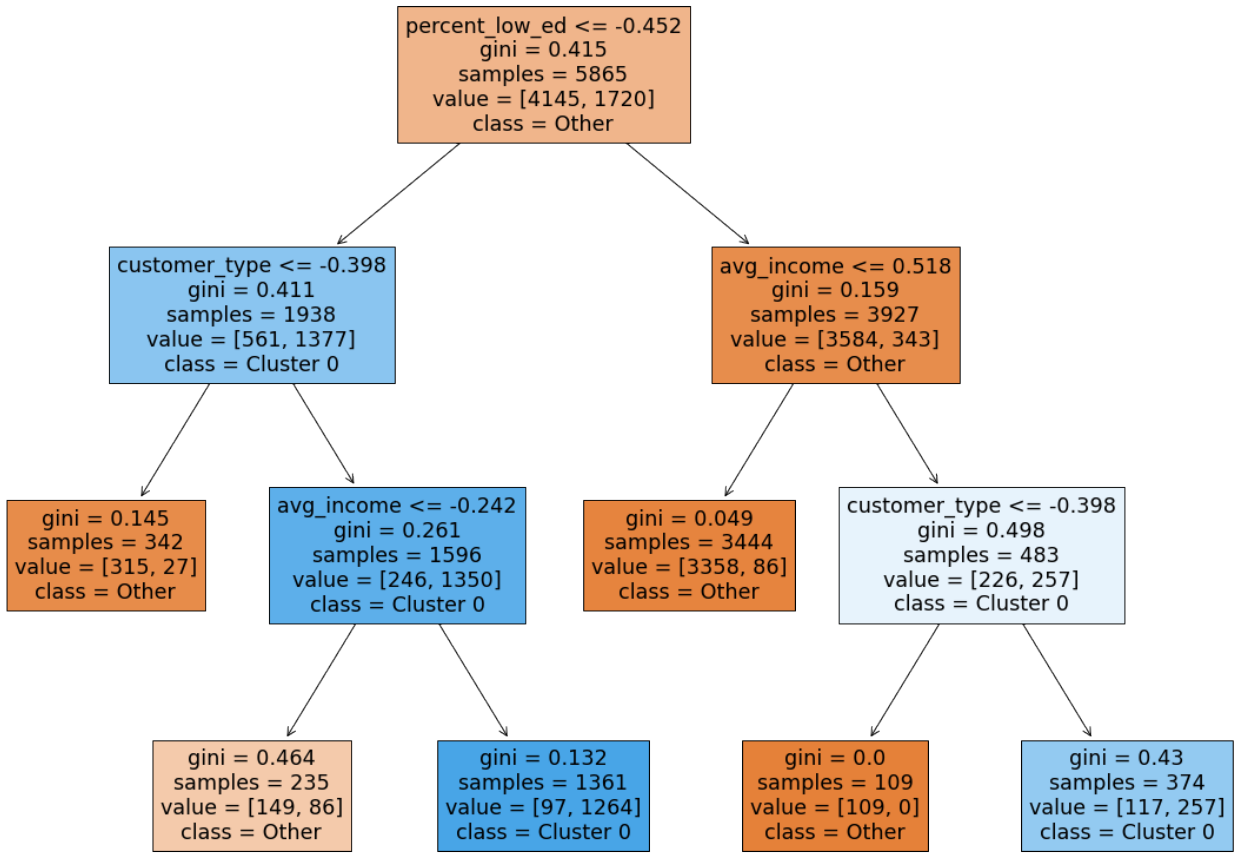

Figure 3: Decision tree for predicting Cluster 0.

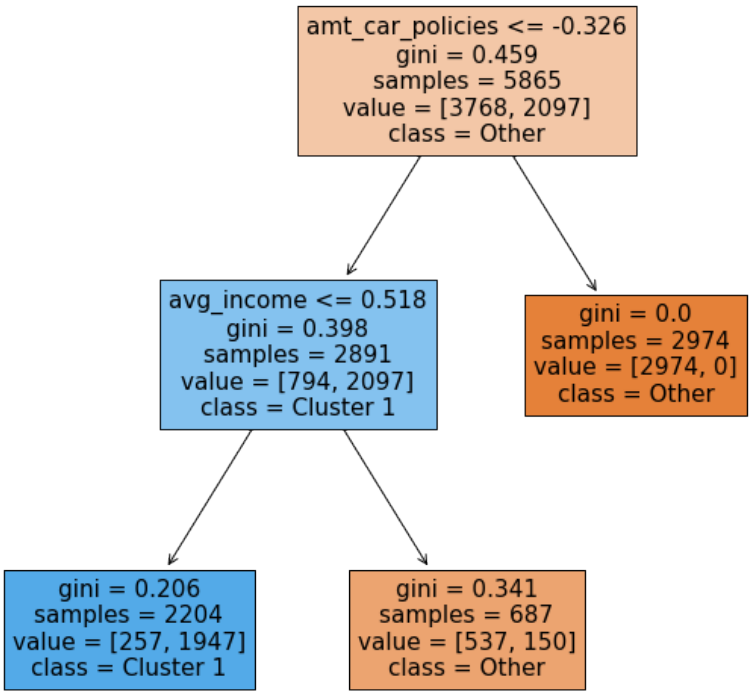

Figure 4: Decision tree for predicting Cluster 1.

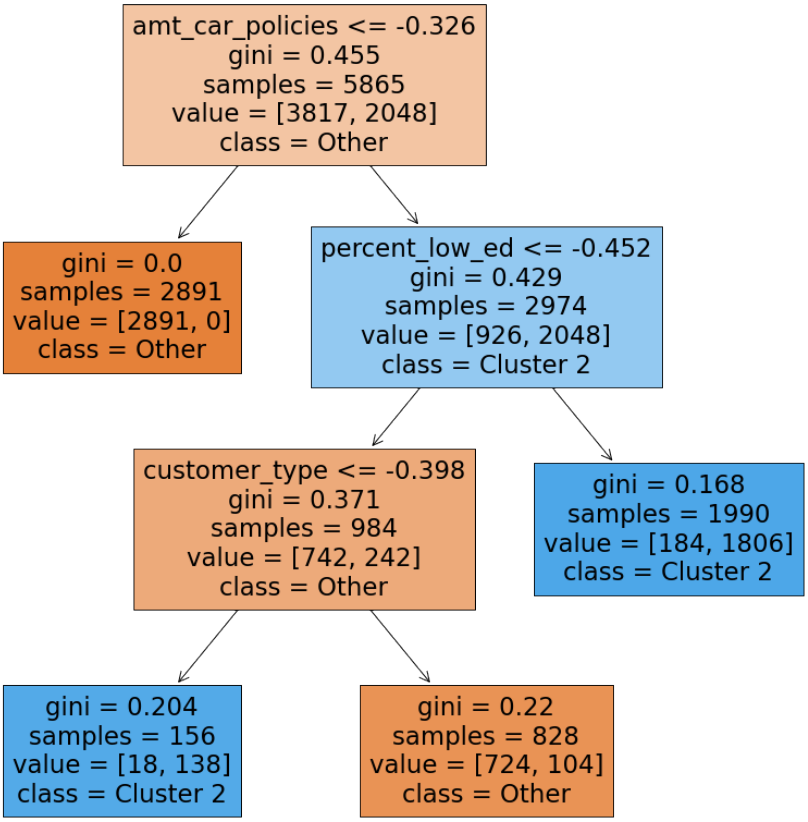

Figure 5: Decision tree for predicting Cluster 2.

References

P. van der Putten and M. van Someren (eds) . CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22, 2000.

Ram Seshadri. featurewiz. [Online]. 2020. Available: https://github.com/AutoViML/featurewiz

UCI Machine Learning. “Caravan Insurance Challenge,” 2017. Retrieved from https://www.kaggle.com/datasets/uciml/caravan-insurance-challenge. Accessed March 14, 2022.