Predicting the Purchase of Caravan Insurance Using Random Forest Classifier

By Mario Gearica, Justin Bouchard, and John Hendricks

Purpose

There are many reasons why purchasing prediction is valuable. The ability to predict consumer behavior can improve marketing campaigns, because businesses can then focus their resources towards marketing to those most likely to purchase their product. Machine learning has already been applied to marketing by top businesses. For example, JP Morgan has used machine learning to create emails and marketing advertisements to specific customers (Forbes).

Our goal was to build a machine learning model that can accurately and reliably predict whether someone in a given postal code will purchase caravan insurance. In addition, we aimed to build a model that is interpretable, meaning the model will provide insights as to what factors influence the purchasing of insurance. The following sections describe how we predicted caravan insurance and the results we achieved.

Nature of the Dataset

The dataset was originally provided at the CoIL Challenge 2000 data mining competition (CoIL Challenge 2000 Website). Each row (10,947 in total) represents a different postal code, and each column (87 in total) indicates some sort of demographic information about that area, including average household size, income levels, and types of insurance policies commonly purchased in that area.

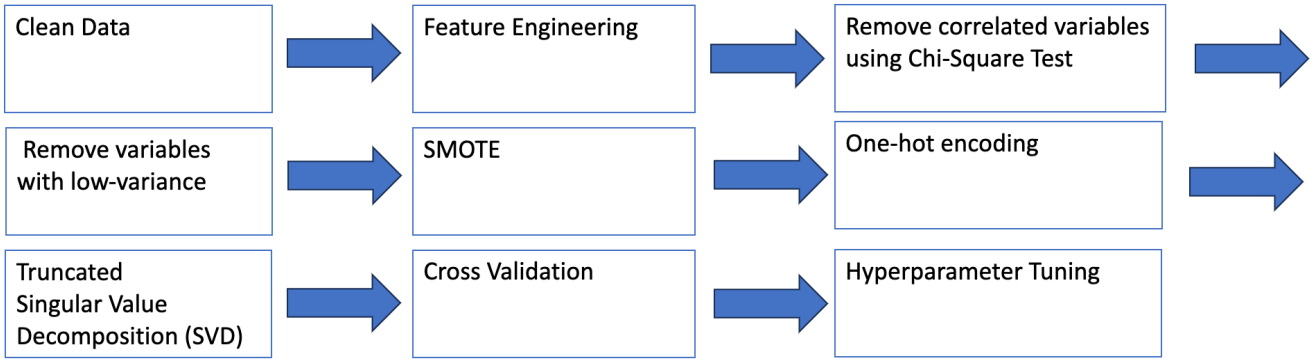
In the data dictionary attached, there are 4 keys : L1, L2, L3, and L4. Each key contains key-value pairs mapping the integer category to the respective bin. For example, variables in the L3 category have a value of 0 to 9, where 0 is 0%, 1 is 1-10%, 2 is 11-23%, etc. The vast majority of variables are therefore categorical, though they are represented by integers.

After briefly looking through the data, we hypothesized that income level and types of insurances bought would be strong indicators of caravan insurance. These predictions are intuitive since 1) income will affect what people purchase 2) if someone tends to insure most of their property, then they will likely insure their caravan as well.

Methodology

Figure 1 shows the workflow used to build the classifier. Preparing the data for the model made up the majority of the steps and effort made. Each step will be summarized in the following sections.

Figure 1 : Machine Learning Workflow for Predicting Caravan Insurance



Data Cleaning

The data contained no null values and no duplicate columns. However, the categorical variables were assigned numbers in an integer format. Initially, we created a correlation matrix with these variables thinking that the Pandas corr function would find correlations of categorical variables. However, the corr function treated these integers simply as integers. When these categories were converted to strings, the correlation matrix no longer included these variables (**Figure 2**).

Feature Engineering:

One effective feature we added was the number of insurance plans purchased ("num_policies"). We counted 1 for each type of insurance bought at that postal code, and used the total count as the new feature. The correlation matrix in **Figure 2** shows there is a small correlation between our engineered variable and the target variable. **Figure 3** shows this distribution of the variable.

Figure 2

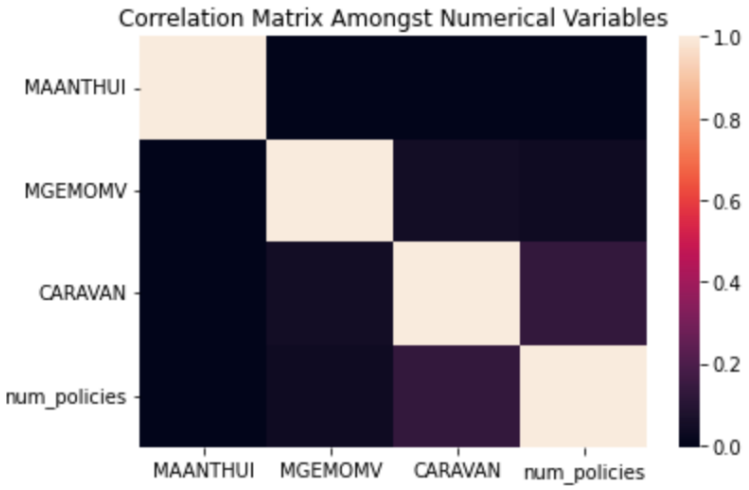
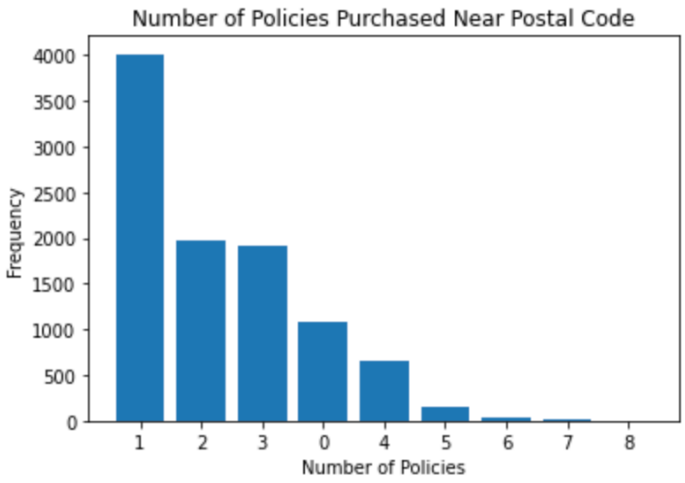


Figure 3



Overall, it was difficult to add new features due to the variables mostly being categorical. We tried adding up the number of contributions from all the policies, as well as creating a new category indicating if there were any high contributions to any policy. These variables did not improve the model, and were therefore discarded.

In many ways, feature engineering is an art form, and therefore with more creativity from the team another effective feature could have been created. The impact of the “num_policies” feature will be analyzed in the later sections.

Removal of Correlated Variables

It is important to remove a feature if it is highly correlated with another feature. For categorical variables, it is not as straightforward to determine correlation as it is with continuous variables. The method used in this analysis was the chi-square test. This statistical test compares expected and observed frequencies among categories and thereby determines statistical significance of their distribution.

Testing all the variables for correlations was the most computationally intensive step of the enitre machine learning workflow. With 84 categorical variables, each possible combination was tested using the chi-square test, resulting in roughly 6,000 tests computed. The time complexity is O(2^n), which is not scalable. **Figure 4** shows examples of the output from the chi-square tests. Many of the tests were well below the normal cut-off value of 0.05.

Figure 4 : P-values from Chi-Square Tests

```
...
 ('MFWEKIND', 'MOPLHOOG', 6.523583792311154e-150),
 ('MFWEKIND', 'MOPLMIDD', 0.0),
 ('MFWEKIND', 'MOPLLAAG', 3.6781365820948257e-268),
 ('MFWEKIND', 'MBERHOOG', 1.2811504477299813e-184),
 ...]
```

This repeated testing led to a common problem in statistics - the more statistical tests you run, the more likely you will find a statistically significant result. To address this issue, we used the Bernoulli technique, where you divide the regular cut-off point (0.05) by the number of tests that you run, which was roughly 6000. This leads to a much more stringent cut-off for statistical significance.

Variables were removed until no more of such relationships were present in the data. A total of 68 of the 87 categories were removed.

Removal of Variables with Low-Variance

Number of surfboard policies (AZEILPL) is the same category 99.91% of the time. This and other such variables were removed from the dataset. This idea was originally from another Kaggle user working on this dataset: Kaggle.

Figure 5 : Variable with Low-Variance

```
AZEILPL
0    0.999084
1    0.000916
Name: AZEILPL, dtype: float64
```

Remaining Variables

The following list details the remaining features after data preprocessing :

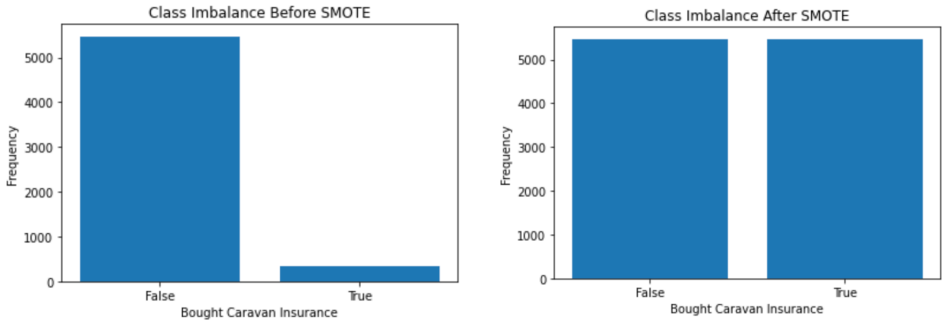
- Number of houses (1 – 10)
- Avg size household (1 – 6)
- Total number of policies (engineered feature)
- Income 30-$45,000, 75-$122, 000, and > $123,000
- Contribution to bicycle policies, motorcycle/scooter policies
- Number of life insurances, Number of bicycle policies
- Number of fire policies, car policies, and tractor policies

This is a total of 13 features. Many of these policies involve a vehicle, which shows promise since we are predicting caravan insurance. It is surprising that income levels were not removed, since it would be expected that these variables are correlated with one another. We currently do not have an explanation why all three of these income variables made it through the chi-square testing instead of just one.

Class Imbalance

Class imbalance is a common problem in classification problems. In the case of this dataset, the vast majority of observations were negative for the purchase of caravan insurance. To improve the quality of the dataset, SMOTE was applied to the data. SMOTE is an oversampling technique that generates synthetic data for the minority class. **Figure 6** shows the class distribution before and after SMOTE. Note that our sample size has almost doubled using this technique, which could have a negative effect on the variance of the data.

Figure 6 : Distribution of Target Variable Before and After SMOTE



One-hot encoding

One-hot encoding was used to transform the categorical variables into a format that machine learning models can read. However, since each variable had 10 categories or more, one-hot encoding resulted in a sparse matrix with over 6 thousand columns filled mostly with zeros. This result is an example of the “Curse of Dimensionality”, where most of the data matrix is empty space. To address this issue, truncated singular value decomposition (SVD) was applied.

SVD

Truncated singular value decomposition reduces the data matrix to an m x n matrix, where m is the number of observations and m is the number of components (Analytics India Magazine). Truncated SVD is like principal components analysis in that it is a data reduction technique, but this technique outperforms PCA when applied to sparse data (Analytics India Magazine).

To determine the correct number of features, a for loop was made such that each model was tested on the data with varying amounts of components. The resulting classification reports decreased beyond 3 components, so the use of 2 components was decided upon.

Models

Cross validation with 5 folds was applied to the following models : Logistic Regression, Random Forest Classifier, ADA Boost, and XG Boost. Random Forest showed the best results with a weighted f1-score of 0.89. Our main issue with Sci-kit Learn's cross_val method was that we could not specify that we wanted the scorer to measure the positive instance of the f1-score. Instead, the model defaulted to an f1-score measuring the negative instance. We chose to use the weighted f1-score that both types of F1 scores were taken into account.

Hyperparameter tuning

Hyperparameter tuning is where you tune specific parameters involved in the learning process of the algorithm. SInce random forest performed the best, its max depth and min_num_samples were optimized. We created a nested for loop where the first loop ran through various max depths and the second for loop ran through various sample numbers for tree splitting. Varying max depth had the most effect on f1-score. A max depth of 3 and a minimum number of samples of 2 was the best performing classifier with an F1-score of 0.19 for the positive instance.

Results:

Figure 7 shows the classification report for the tuned random forest model. The f1-score for the positive instance (1) is 0.19. The f1-score ranges from 0 to 1, with 1 being the highest, so there is much room for improvement. The precision is 11%, meaning that our model accurately predicts the purchase of caravan insurance 11% of the time. The recall is 54%, which means our model is picking up on the majority of the positive instances.

Figure 7 : Classification Report

```
# final model
rf = RandomForestClassifier(max_depth=3, min_samples_split=2).fit(transformed, y_train)
y_pred = rf.predict(transformed_)
print(classification_report(y_test, y_pred))
```

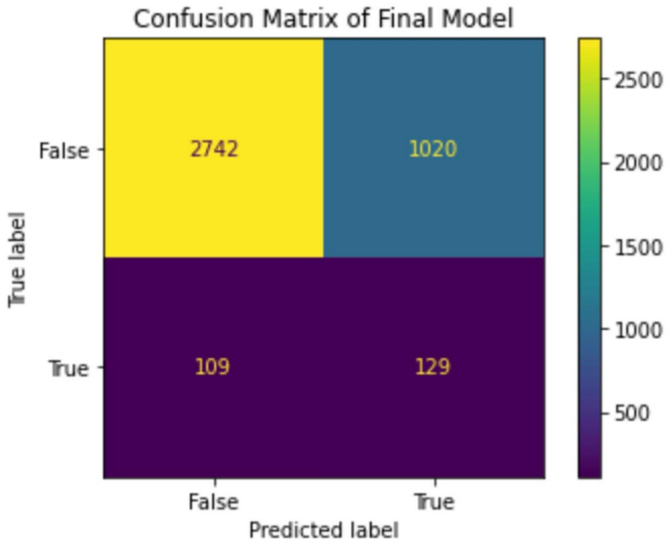
```
              precision    recall  f1-score   support

           0       0.96      0.74      0.83      3762
           1       0.11      0.54      0.19       238

    accuracy                           0.73      4000
   macro avg       0.54      0.64      0.51      4000
weighted avg       0.91      0.73      0.80      4000
```

Figure 8 shows the confusion matrix for the same model. The top-right quadrant shows that the model has many false positives (low precision). However, the model does capture the majority of the true positives (high recall) as mentioned in previously.

Figure 8



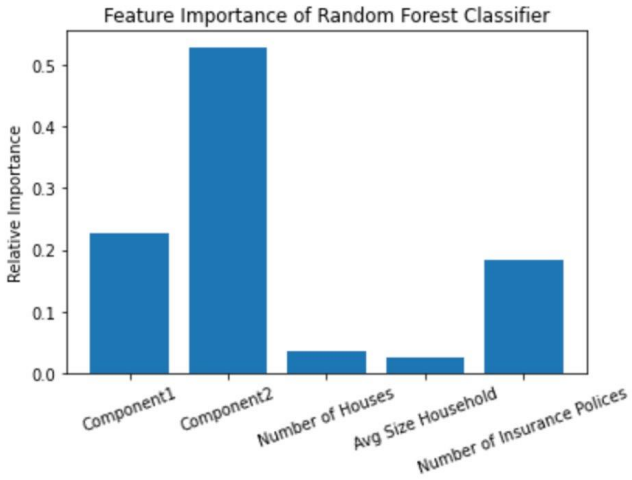
In terms of marketing strategy, it is acceptable to have false positives because the default goal of marketing is to reach as many potential customers as possible. There are no adverse consequences to false positives in marketing, unlike other files like medicine, where a false positive could be an expensive and possibly life-threatening mistake.

Nevertheless, the model needs significant improvement to be applied to a business workflow.

Feature Importance

Figure 9 shows the relative importance of each feature in the data. As hypothesized, the number of insurance policies has an impact on the decision the classifier comes to. Component 2 has the highest level of importance in predicting the purchase of caravan insurance.
The issue is that Components 2 is truncated by SVD, so it is difficult to interpret this result. Since truncated SVD is essentially a combination of linear algebra methods, we hypothesize that linear algebra could be used to determine the meaning behind these components and thus give further explanation to what the model is predicting.

Figure 9



Conclusion

Our goal was to build a machine learning model by which we would identify actionable insights and profit-maximizing opportunities with regards to improving corporate marketing and sales-incentive campaigns. Using k-fold cross validation, the best model was determined to be Random Forest Classifier. After hyperparameter tuning, the final model had 11% precision and 54% recall for the positive class, with a f1-score of 0.19. The primary features influencing the model were 1) income distribution and 2) the purchase of other forms of insurance. Next steps include adding more features, testing other resampling methods, and testing a neural network on the data. Despite the amount of improvement left to be made in terms of precision and recall, this machine learning analysis demonstrates various principles and techniques in machine learning, and therefore serves as a valuable educational tool for practicing and learning the fundamentals of data science.