DNSC 6279 – Assignment 2

Download the Claims Prediction Challenge data from Kaggle or from my dropbox.

The data is available at these locations:

Kaggle: https://www.kaggle.com/c/ClaimPredictionChallenge/data
Dropbox: https://www.dropbox.com/sh/l4827tbx4cwl1t2/AADYQs32AzumeL-GFaZPvQF1a?dl=0

(You can read more about this data on the Kaggle competition data page linked above.)

1.) **(2 pts.) Decompress the contest data and read it into Python, R, or SAS. Your choice.**

The 7zipped files can be uncompressed into normal CSV files using 7Zip (http://www.7-zip.org/). The uncompressed files will require about 3.5 GB of disk space.

Once you have read these data sets into your tool of choice, either as a data frame or a data set, how many rows and columns does train_set have? How many rows and columns does test_set have?

2.) **(4 pts.) Engineer a feature for past claims.**

Create two variables in both the training and test sets that indicate the total number of past claims per household and the total claim amount per household. The target for the contest was claim_amount. One major indicator of whether a claim will be made on a vehicle is whether the household that owns the vehicle has made a claim in the recent past. (Think about teenage or elderly drivers.) Note that households can have multiple vehicles.

In train_set how many past claims does household 4719940 have? Total claim amount?

In test_set how many past claims does household 3372413 have? Total claim amount?

3.) **(1 pt.) Identify problematic, high-cardinality categorical variable levels.**

Blind_model and Blind_submodel are problematic. They contain levels in the test data that are not present in the training data. Since the model will not see these levels during training, it will struggle to use them when making predictions.

How many levels of the categorical variable blind_model exist in the test_set but not in the train_set?

(Decision tree models can be used in this scenario due to their ability to use surrogate rules to make decisions about unseen categorical variable levels, such as using the customer's state when a splitting rule is not available for their zip code. Such problematic values are often times simply replaced using common-sense business rules too.)

4.) **(2 pts.) Use Weight of Evidence to fix problematic, high-cardinality categorical variables.**

The variables blind_model and blind_submodel are problematic. They contain levels in the test data that are not in the training data and have too many levels for dependably training a model. However they provide a great deal of information – of course there is a link between the model of car someone drives and the probability that they will file an insurance claim. These high-cardinality categorical variables can be converted to interval variables using several techniques, including Weight of Evidence (WOE). In this case, WoE is defined as:

$$100\ x\ \ln\left(\frac{frequency\ of\ event}{frequency\ of\ nonevent}\right)$$

Where

$$frequency\ of\ event = \frac{number\ of\ nonzero\ value\ claims\ in\ level}{total\ number\ of\ nonzero\ value\ claims}$$

$$frequency\ of\ nonevent = \frac{number\ of\ zero\ value\ claims\ in\ level}{total\ number\ of\ zero\ value\ claims}$$

What is the WoE value for blind_model = X.45? and blind_submodel = X.45.2?

(Typically WoE would be calculated for every level of blind_model and blind_submodel, and each categorical level in the training and test set would be replaced by the numeric WoE value. That is not necessary for this assignment.)

5.) **(1 pt.) Create a balanced training set using oversampling.**

The number of nonzero claims in the training set is very small. Such rare events can lead to problems training a model dependably. One common and simple approach to address rare events is oversampling.

Create a sample of the training data with every record containing a nonzero claim and an equal number of randomly selected records containing zero value claims. How many records are in this sample?