Accurate Campaign Targeting Using Classification Algorithms

Jieming Wei Sharon Zhang

Introduction

Many organizations prospect for loyal supporters and donors by sending direct mail appeals. This is an effective way to build a large base, but can be very expensive and have a low efficiency. Eliminating likely non-donors is the key to running an efficient Prospecting program and ultimately to pursuing the mission of the organization, especially when there's limited funding (which is common in most campaigns).

The raw dataset, with 16 mixed (categorical and numeric) features, is from Kaggle website – Raising Money to Fund an organizational Mission. We built a binary classification model to help organizations to build their donor bases in the most efficient way possible, by applying machine learning techniques, including Multinomial Logistic Regression, Support Vector Machines, Linear Discriminant Analysis. We aimed to help them separate the likely donors from the non-likely donors and target at the most receptive donors by sending direct mail campaigns to the likely donors.

Data Preparation: feature selection, training set and test set selection

The original dataset is 13-dimensional with 1048575 observations on demographic features of prospects and campaign attributes.

- Demographic: zip code, mail code, company type.
- Prospect attributes: income level, job, group.
- Campaign attributes: campaign causes, campaign phases.
- Result : success/failure

In order to serve the purpose of this study, we adapted the dataset, reducing it to a 10-dimensional dataset to keep the relevant features and selected training set and test set:

Feature selection

We selected features based on their relevance to the result of the campaign, estimated by the correlation of a given feature to the result. We selected 7 features with relatively large correlation with the result. We found that donor income level is the most correlated variable.

Table 1: Feature analysis

company type	project type	cause1	cause2	phase	zip	donor income	result
-0.016	-0.0008	-0.0765	-0.0416	0.0149	0.071	**0.248**	1

Original dataset:

13-dimensional with 1048575 observations

Features:

Project ID, Mail ID, Mail Code ID, Prospect ID, Zip, Zip4, Vector major, Vector Minor, Package ID, Phase, Database ID, JobID, Group, ID

Processed dataset:

7 dimensional. Training set: 300 observations. Test set: 100 observations.

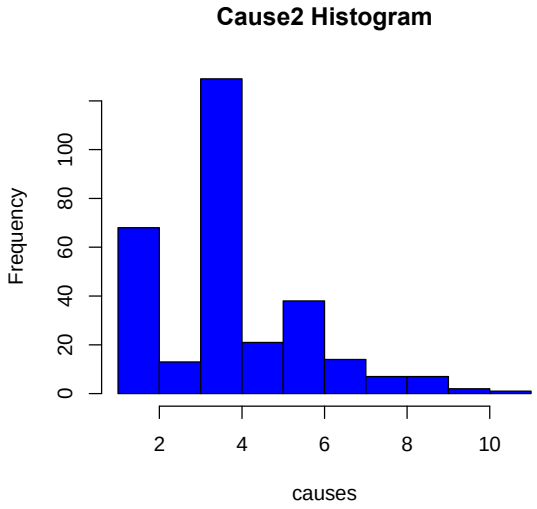
Features:

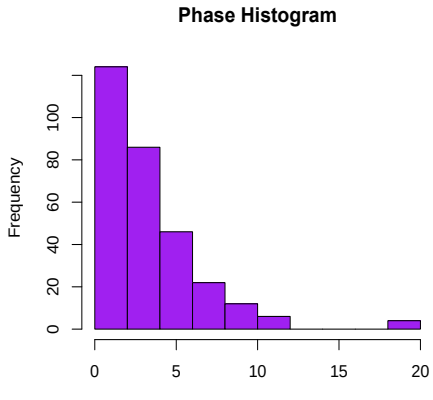
- Company type (Campaign Organization Type),
- Project type,
- Cause 1, (indicating whether the campaign is for profit or non-profit)
- Cause 2, (further classifying causes into policy, tax, PAC, CNP, congress, MEM.)
- Phase,
- Zip,
- Candidate Income,
- *result. The distribution of these variables are displayed below.

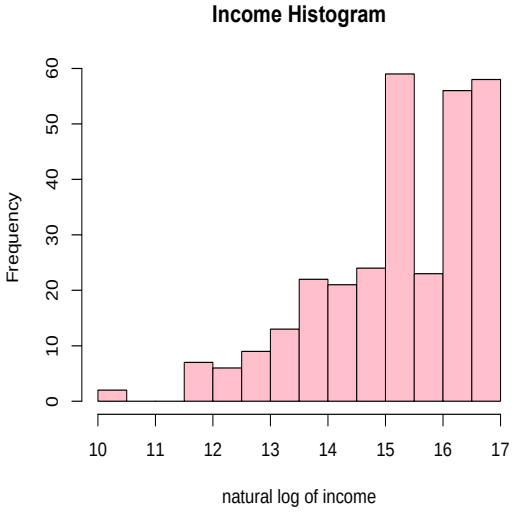
Where *result is a binary value indicating good/bad prospects. It is where we'll base our training and gauge the prediction.

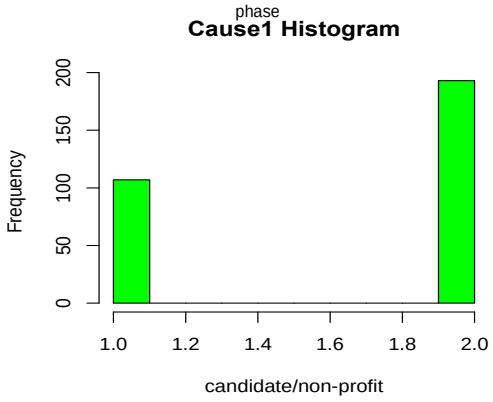
Distribution visualization

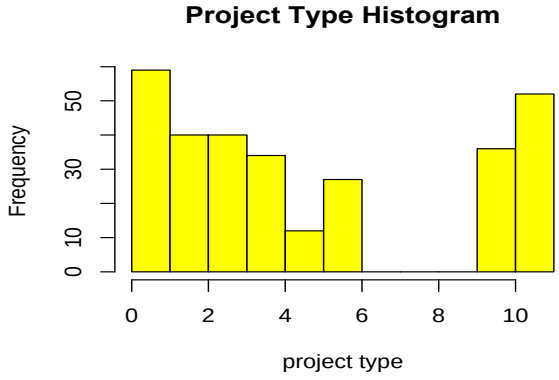
We generated histogram to understand the distributions of the features. As candidate income, the most relevant feature, has large variance, we performed log transformation to visualize its distribution, as well as its

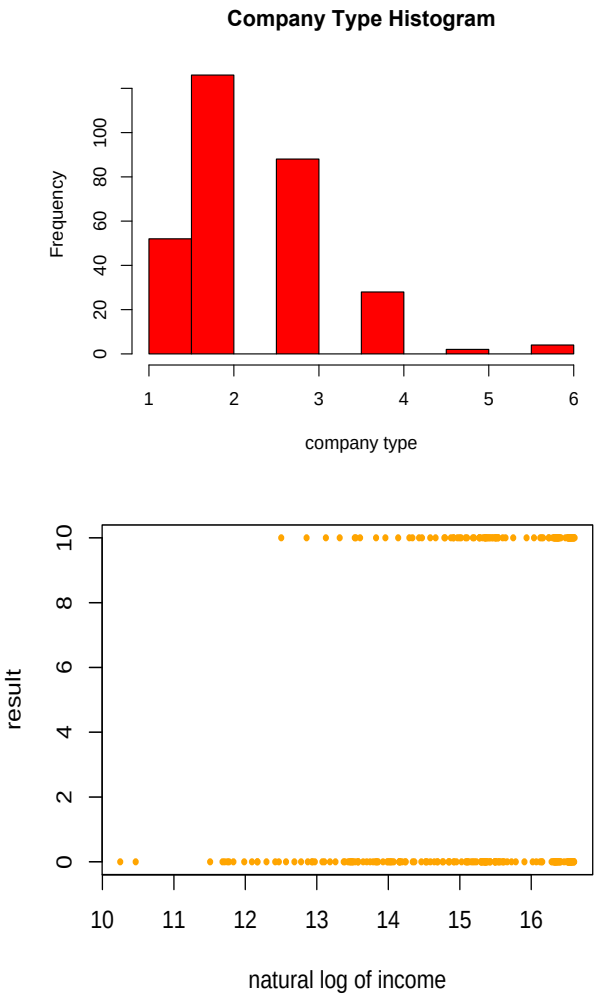












distribution with respect to result.

Training set and Test set Selection

We created a training set by randomly selected 300 observations from the original dataset, and 100 observations as the test set. And replaced categorical variables with numbers.

- Cause 1: 1, 2 are replaced for candidate/non-profit.
- Cause 2: 1-11 are replaced for congress, CNP, tax, policy, DEF, senate, PAC, MIL, MEM, IMM, REL.

Methods

We tried the following approaches to classify mailings into two classes, namely donors who are likely to donate and those who are not: Multivariate Regression, SVM, Decision Tree, Neural Network, and Random Forest. Training errors , test errors, false positive error rates, false negative error rates are calculated for each model.

	Training error	Overall test error	**False positive**	False negative
MVR	0	0.31	**0.25**	0.06
SVM	0.323	0.54	**0.18**	0.36
Neural Nets	--	0.378	**0.174**	0.204
Random Forest	0.003	0.47	**0.23**	0.24

We noticed that test errors for most models are relatively high, with MVR and Random Forest models having high variance problems, and SVM having high bias problems. False positive errors are high for most models, with Neural Net model having the least false positive error. We want to minimize false positive error so as to save costs for campaign companies.

Algorithm Selection

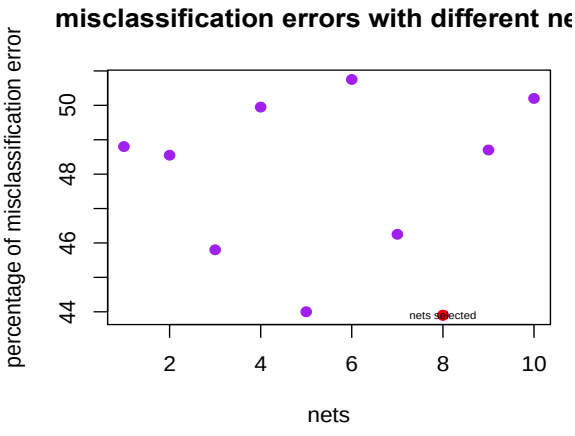
From the previous initial fitting with different algorithms, we found that the dataset is a little messy and irregular in terms of patterns – there are some random processes in which a prospect would end up donating money or not. It is a character of this dataset, while it is not uncommon in real world datasets.
When datasets have this degree of randomness, it is hard to lower the overall error ratio, however, we can still distinguish false positive and false negative, and specifically lower the error type that yields better profit, saves more efforts, etc.

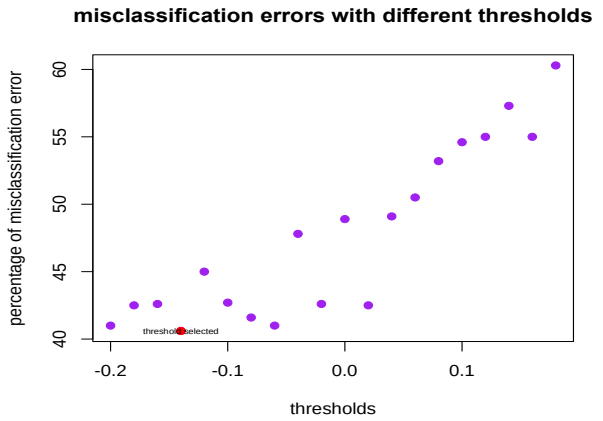
whatever makes more sense. In this case, we want to lower false positive error. It is because when a prospect is mistakenly considered "will donate", then time and money resources would be spent and wasted.

We choose to use Neural Network for this experiment, since it has the lowest false positive rate. There are three steps – first choose the number of neural nets and the threshold for the structural model, then choose the optimal regularization parameter under this optimized structural model, and lastly choose the threshold for false positive minimization.

1) Pick the number of neural nets and threshold for the structural model

We did a looping experiment with 10 nets, with 20 threshold values ranging from -0.2 to 0.2. For each value pair we calculate an overall estimation of misclassification, and get a 20 × 10 matrix. Applying averages to rows and columns, we choose the lowest average in both column and row to be the number of nets and threshold value. The result of applying average is as below:

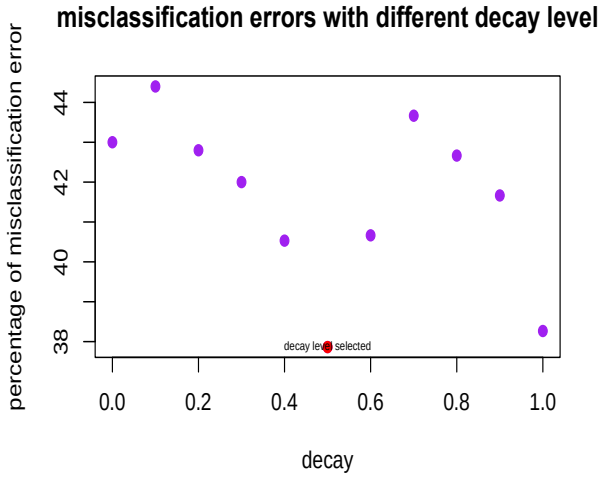




Thus, we use -0.14 as threshold and 8 layers of hidden nets for our structural model.

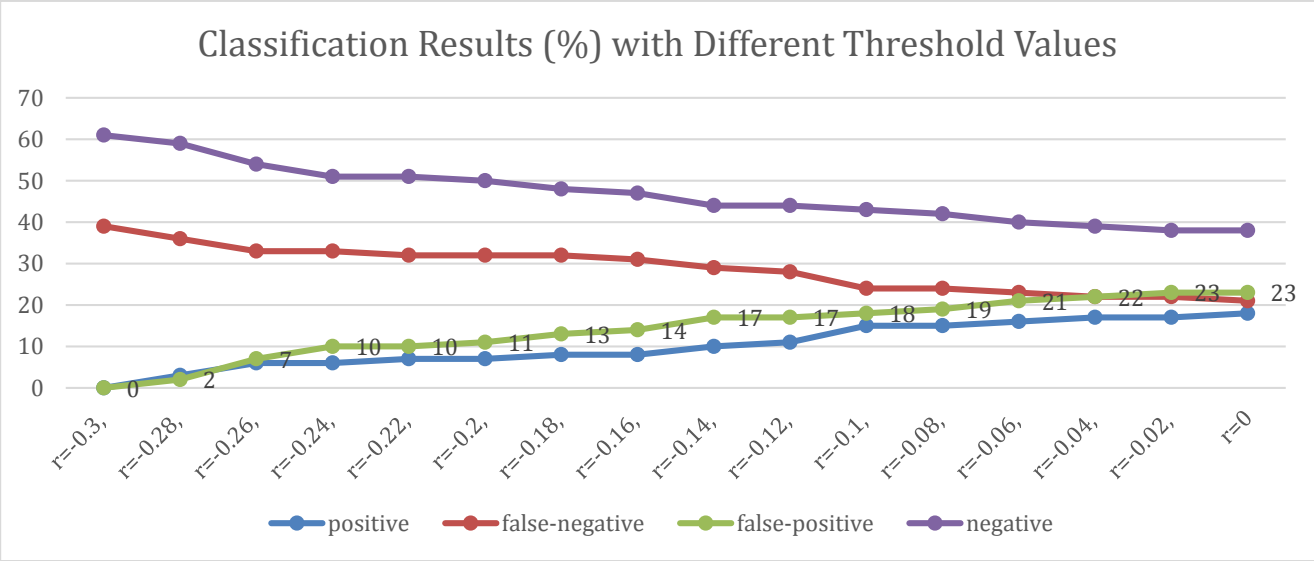
2) Determine the optimal regularization under the structural model chosen

This step we choose the optimal regularization (weight decay for parameters 0, 0.1, … , 1) for the structural model found above by averaging our estimators of the overall misclassification error on the test set. We calculate the average with over 10 runs with different starting values. From this tuning process we found that decay parameter 0.5 yields the lowest overall misclassification.



3) Determine threshold to minimize false positive misclassifications

A way to control the false positive misclassified donors to be less is to choose a threshold of being a donor or non-donor.



When there is a higher "bar" for classifying prospects to be a donor, we filter the prospects with most certainty of being a donor.

We tried 16 different threshold values and drew the result chart with four different categories – positive, false negative, false positive, negative. When threshold is -0.3, every prospect is classified as non-donators, thus the false positive error is 0. When the threshold increases, false positive errors increase while false negative error decrease.

The question now is to find a balanced point of two types of errors so that the overall campaign is cost effective. We use the following formula:

$$\textbf{\textit{Net Collected}}\ (\$) = \textbf{\textit{Donation}}\ (\$) * \textbf{\textit{Donors}}\ (\boldsymbol{M}) - \textbf{\textit{Package Cost}}\ (\$) * \textbf{\textit{Packages}}\ (\boldsymbol{N})$$

Where Donation ($) and Package Cost($) are assumptions, Donors (M) corresponds to the number of positive occurrences, and Packages Sent (N) corresponds to the sum of positive and false-positive occurrences since the model predicts those prospects will donate. When we use the assumptions Donation = $50 and Package Cost = $20, we observe the following Net Collected amount trend in chart on the right.

Thus, the most cost-effective threshold is r=-0.1, with false-positive rate 18%.