## **Direct-Mail Fundraising**

**NOTES:**

**Question Statement:**

*Attached Fundraising.csv* and *FutureFundraising.csv* are the datasets used for this Case

Study.

**Background:**

*A national veterans’ organization wishes to develop a predictive model to improve the cost-effectiveness of their direct marketing campaign. The organization, with its in-house database of over 13 million donors, is one of the largest direct-mail fundraisers in the United States. According to their recent mailing records, the overall response rate is 5.1%. Out of those who responded (donated), the average donation is $13.00. Each mailing, which includes a gift of personalized address labels and assortments of cards and envelopes, costs $0.68 to produce and send. Using these facts, we take a sample of this dataset to develop a classification model that can effectively capture donors so that the expected net profit is maximized.*

*Now, Weighted sampling is used, under-representing the non-responders so that the sample has equal numbers of donors and non-donors.*

**Table: Description of Variables for the Fundraising Dataset**

|  |  |
| --- | --- |
| Variable | Description |
|  |  |
| ***ZIP*** | *Zip code group (zip codes were grouped into five*  *groups;* |
|  | *1 = the potential donor belongs to this zip group.)* |
|  | *00000–19999 ⇒ zipconvert\_1* |
|  | *20000–39999 ⇒ zipconvert\_2* |
|  | *40000–59999 ⇒ zipconvert\_3* |
|  | *60000–79999 ⇒ zipconvert\_4* |
|  | *80000–99999 ⇒ zipconvert\_5* |
| ***HOMEOWNER*** | *1 = homeowner, 0 = not a homeowner* |
| ***NUMCHILD*** | *Number of children* |
| ***INCOME*** | *Household income* |
| ***GENDER*** | *0 = male, 1 = female* |
| ***WEALTH*** | *Wealth rating uses median family income and*  *Population statistics from each area to index relative wealth within each state.*  *The segments are denoted 0 to 9, with 9 being the highest-wealth group and zero the lowest. Each rating has a different meaning within each state.* |
| ***HV*** | *Average home value in potential donor’s neighbourhood in hundreds of dollars.* |
| ***ICmed*** | *Median family income in potential donor’s neighbourhood in hundreds of dollars.* |
| ***ICavg*** | *Average family income in potential donor’s neighbourhood in hundreds.* |
| ***IC15*** | *Percent earning less than $15K in potential donor’s neighbourhood.* |
| ***NUMPROM*** | *Lifetime number of promotions received to date.* |
| ***RAMNTALL*** | *Dollar amount of lifetime gifts to date.* |
| ***MAXRAMNT*** | *Dollar amount of largest gift to date.* |
| ***LASTGIFT*** | *Dollar amount of most recent gift.* |
| ***TOTALMONTHS*** | *Number of months from last donation to July 1998 (the last time the case was updated).* |
| ***TIMELAG*** | *Number of months between first and second gift.* |
| ***AVGGIFT*** | *Average dollar amount of gifts to date* |
| ***TARGET\_B*** | *Outcome variable: binary indicator for response 1 = donor, 0 = non-donor* |
| ***TARGET\_D*** | *Outcome variable: donation amount (in dollars). We will NOT be using this variable for this case.* |

**Some Code and Explanation from the Code:**

**Performing Logistic Regression**

logit\_reg = LogisticRegression(penalty="l2", C=1e42, solver='liblinear')

logit\_reg.fit(train\_X, train\_y)

*LogisticRegression(penalty="l2", C=1e42, solver='liblinear'): This line initializes a logistic regression model with the specified parameters:*

* *penalty="l2": Indicates that the regularization penalty used is L2 regularization, also known as Ridge regularization.*
* *C=1e42: Specifies the inverse of regularization strength. A larger C value indicates weaker regularization, effectively reducing the impact of regularization on the model.*
* *solver='liblinear': Indicates the optimization algorithm used to solve the logistic regression problem. 'liblinear' is a solver suitable for small to medium-sized datasets and supports both L1 and L2 regularization.*

*logit\_reg.fit(train\_X, train\_y): This line fits the logistic regression model to the training data:*

* *train\_X: Represents the independent variables (features) of the training dataset.*
* *train\_y: Represents the dependent variable (target variable) of the training dataset.*
* *The fit() function trains the logistic regression model by adjusting the coefficients to minimize the logistic loss function based on the provided training data*

*print('intercept ', logit\_reg.intercept\_[0])*

*The code print('intercept ', logit\_reg.intercept\_[0]) prints the intercept value of a logistic regression model. In logistic regression, the intercept (also known as the bias) represents the expected outcome when all predictor variables are set to zero.*

* *logit\_reg.intercept\_: This is an attribute of the logistic regression model object (logit\_reg) that holds the intercept value(s) calculated during model training.*
* *[0]: In logistic regression models fitted with scikit-learn, the intercept is typically stored as an array. The [0] indexing is used to access the first (and usually only) element of this array.*
* *print('intercept ', logit\_reg.intercept\_[0]): This line prints the word "intercept" followed by the actual intercept value of the logistic regression model.*
* *The intercept value is important because it determines the baseline probability of the positive class in logistic regression. It shifts the logistic function along the vertical axis and influences the decision boundary of the model.*

*logit\_reg\_pred = logit\_reg.predict(valid\_X)*

*logit\_reg\_proba = logit\_reg.predict\_proba(valid\_X)*

*logit\_result = pd.DataFrame({'actual': valid\_y,*

*'p(0)': [p[0] for p in logit\_reg\_proba],*

*'p(1)': [p[1] for p in logit\_reg\_proba],*

*'predicted': logit\_reg\_pred })*

*This code generates a DataFrame named logit\_result containing various information related to logistic regression predictions on the validation dataset (valid\_X and valid\_y). Here's a breakdown of what each part of the code does:*

*logit\_reg\_pred = logit\_reg.predict(valid\_X): This line predicts the target values (logit\_reg\_pred) using the logistic regression model (logit\_reg) on the validation features (valid\_X).*

*logit\_reg\_proba = logit\_reg.predict\_proba(valid\_X): This line predicts the class probabilities (logit\_reg\_proba) using the logistic regression model (logit\_reg) on the validation features (valid\_X). For each sample, it provides the probability of belonging to each class.*

*logit\_result = pd.DataFrame({'actual': valid\_y, 'p(0)': [p[0] for p in logit\_reg\_proba], 'p(1)': [p[1] for p in logit\_reg\_proba], 'predicted': logit\_reg\_pred }): This line creates a DataFrame named logit\_result containing four columns:*

*'actual': This column contains the actual target values from the validation dataset (valid\_y).*

*'p(0)' and 'p(1)': These columns contain the predicted probabilities of belonging to class 0 and class 1, respectively, for each sample in the validation dataset. These probabilities are extracted from the logit\_reg\_proba array.*

*'predicted': This column contains the predicted target values (logit\_reg\_pred) generated by the logistic regression model.*

*Overall, logit\_result provides a comprehensive view of the model's predictions on the validation dataset, including the actual target values, predicted probabilities, and predicted target values. This DataFrame can be further analyzed to evaluate the performance of the logistic regression model.*

**Objective**

*The purpose of this project is to build a classification model in order to improve the cost effectiveness of a national veteran’s organization’s direct marketing campaign. This model will help by predicting which individuals will be more likely to donate to the organization as opposed to donors who will not donate.*

**Steps Taken to achieve result**

*Step 1—Partitioning: Partition the dataset into 60% training and 40% validation (setting the seed to 12345).*

*Step 2—Model Building: Following steps to build, evaluate, and chose a model.*

*Step 3—Drawing cumulative gains curves: Drawing the different models’ cumulative gains curves for the validation set in a single plot.*

*Step 4—Selecting best model: From your answer in (2).*

*Step 5—Testing: The file FutureFundraising.csv contains the attributes for future mailing candidates.*

*Step 6—Using your “best” model from Step 2, which of these candidates do you predict as donors and non-donors? List them in descending order of the probability of being a donor. Starting at the top of this sorted list.*

**Conclusions/Recommendations**

*Based on the various models which were run during this project, the best variables which worked best were months\_since\_donate, num\_child, num\_prom, pct\_lt15k, and avg\_fam\_inc with an 60/40 split with no transformations or variable exclusions being done on the model.*

*In summary, the most important predictors for deciding whether an individual will donate to this marketing campaign based on my models are a combination of the 5 variables:*

1. *Holding all other variables constant, the number of months since a person has donated will have a significant effect as to whether an individual will donate to the campaign.*
2. *Holding all other variables constant, the average family income will have a significant effect as to whether an individual will donate to the campaign.*
3. *Holding all other variables constant, the lifetime number of promotions received to date will have a significant effect as to whether an individual will donate to the campaign.*
4. *Holding all other variables constant, the percent earning less than $15K in a potential donor’s neighborhood will have a significant effect as to whether an individual will donate to the campaign.*

1. *Holding all other variables constant, the number of children will have a significant effect as to whether an individual will donate to the campaign.*

*I believe looking at these 5 predictors from the fundraising dataset will help improve the cost effectiveness of the direct marketing campaign’s efforts as well as predict who is more likely to donate to the campaign.*