

Master Thesis

An application of data-driven decision making

Profit maximization for direct marketing campaigns

Submitted in partial fulfillment of the requirements for the degree of Master of Science in
Statistics

Florian Hochstrasser

14.06.2019

Supervisor: Jacques Zuber

Abstract

A profit optimization problem, based on the data distributed for the KDD-Cup 1998, was solved using several different supervised learning algorithms. A complete analysis including data preprocessing, model evaluation and selection and construction of the final estimator was conducted.

During preprocessing, several approaches for imputation and feature selection were explored. A two-stage prediction approach was chosen, separately predicting donation probability and donation amount. Five classification- and four regression-algorithms were evaluated. The best-performing algorithms of both families were then combined for the final prediction.

A python package was developed to automate preprocessing of the data, guaranteeing consistent treatment of learning- and validation data. The package features an extensible architecture for future work on the data set.

In comparison to the winners of the Cup, the developed model ranked 4th. Future work based on the results achieved should focus on feature engineering and alternative imputation strategies.

Contents

1. Introduction	7
1.1. Task Background	7
1.2. Goals and Requirements	8
1.3. Conventions and Notes	8
2. Data	9
2.1. General Structure	9
2.2. Exploratory Data Analysis	9
2.2.1. Data Types	10
2.2.2. Targets	10
2.2.3. Skewness	12
2.2.4. Correlations	13
2.2.5. Donation Patterns	14
3. Experimental Setup and Methods	19
3.1. Tools Used	19
3.2. Data Handling	19
3.3. Data Preprocessing	20
3.3.1. Cleaning	20
3.3.2. Feature Engineering	21
3.3.3. Imputation	21
3.3.4. Feature Selection	23
3.4. Prediction	23
3.4.1. Setup of the Two-Stage Prediction	24
3.4.2. Optimization of α^*	25
3.5. Model Evaluation and -Selection	25
3.5.1. Evaluation	26
3.5.2. Selection	26
3.5.3. Dealing With Imbalanced Data	28
3.5.4. Algorithms	28
4. Results and Discussion	36
4.1. Preprocessing With Package kdd98	36
4.2. Imputation	36
4.3. Feature Selection	37
4.4. Model Evaluation and Selection	38
4.4.1. Classifiers	38
4.4.2. Regressors	42

Contents

4.5. Prediction	44
4.5.1. Prediction of Donation Probability	44
4.5.2. Conditional Prediction of the Donation Amount	45
4.5.3. Profit Optimization	45
4.5.4. Final Prediction	46
5. Conclusions	48
References	49
A. Software	52
A.1. Python Environment	52
A.2. Package kdd98	52
A.2.1. Usage	52
A.2.2. Installation	53
B. KDD Cup Documents	54
B.1. Cup Documentation	54
B.2. Data Set Dictionary	79

List of Figures

2.1. Distribution of the binary target TARGET_B	11
2.2. Distribution of TARGET_D , the donation amount in \$ US (only amounts > 0.0 \$ are shown).	11
2.3. Fisher-Pearson standardized moment coefficient (G1) for all numeric features contained in the dataset. The confidence bound indicates the $\alpha = 5\%$ bound for the skewness of a normal distribution for any given feature.	12
2.4. Least skewed features by G1 (adjusted Fisher-Pearson standardized moment coefficient).	12
2.5. Most skewed features by G1 (adjusted Fisher-Pearson standardized moment coefficient).	13
2.6. Heatmap of feature correlations. Green means positive correlation, magenta means negative correlation. Perfect correlation occurs at 1.0 and -1.0.	14
2.7. Analysis of all-time RFA values by response to current promotion. <i>Recency</i> is the time in months since the last donation, <i>Frequency</i> the average number of donations per year and <i>Amount</i> the average yearly donation amount.	15
2.8. Donation amount for the current promotion against months since last donation. The dot size indicates the number of times an example has donated.	15
2.9. Frequency of donations in the 13-24 months prior to current promotion against amount donated. Frequent donors give smaller amounts.	16
2.10. Geographical distribution of donations by zip code. Point size indicates total donations for a zip code while the hue shows average donation amount.	17
2.11. Average cumulative donation amount per capita by living environment (C = city, U = urban, S = suburban, T = town, R = rural). The more rural, the higher the average donations.	17
2.12. Donation amount for current promotion by living environment and socio-economic status of examples. The violin plot shows the distribution of values similar to a kernel density estimation. Median values are indicated by white dots, the bold regions give the inner quartile range.	18
3.1. Data set use for training and predictions.	20
3.2. Learning process schematic.	25
3.3. Definition of the confusion matrix for a two-class problem.	27
3.4. Neural network topology used. Two hidden layers $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}$ are contained. $\mathbf{b}^{(2)}, \mathbf{b}^{(3)}$ and $\mathbf{b}^{(4)}$ are the bias vectors for the respective layers.	33
3.5. Schematic display of an SVM hyperplane (in black), separating two overlapping classes. The margins are shown around the hyperplane, with support vectors falling on the margins. Misclassifications (examples on the wrong side of the hyperplane) have a total budget for distance from the separating plane. The margins are determined by respecting the budget. Adapted from Friedman, Hastie, and Tibshirani (2001).	34
4.1. Features with most (left) and fewest (right) missing values.	37

List of Figures

4.2. Comparison of recall scores for all classifiers evaluated.	39
4.3. Comparison of ROC-AUC for the evaluated classifiers (left) and PR curves (right) for the classifiers.	40
4.4. Confusion matrices for the 6 classifiers studied.	41
4.5. Coefficient values for GLMnet.	41
4.6. Feature importances determined with the RF classifier. Impurity is measured by Gini importance / mean decrease impurity. Error bars give bootstrap error on 50 repetitions.	42
4.7. Target before transformation (left) and after a Box-Cox transformation (right).	42
4.8. Distribution of (Box-Cox transformed) donation amounts for the four regressors evaluated.	43
4.9. Evaluation metric R^2 for all regression models evaluated. The domain for R^2 is $(-\infty, 1]$	43
4.10. Feature importances for the RF regressor.	44
4.11. Distribution of predicted donation probabilities \hat{y}_b on the learning data set.	44
4.12. Conditionally predicted donation amounts, Box-Cox transformed (left) and on the original scale (right).	45
4.13. Expected profit for a range of α values in $[0, 1]$ with overlayed cubic spline and polynomial function of order 12.	46

1. Introduction

Customer segmentation techniques are used in marketing to identify certain groups of customers in order to produce offers tailored to these groups. Customer segmentation needs a goal to be set. Generally, customer segmentation is performed with the overall goal of increasing the profitability of a business. Retaining profitable customers is one way to achieve that goal. In the case of direct marketing, especially when unit costs (the cost associated with addressing a customer) are significant, employing some customer segmentation technique is highly beneficial in terms of profit.

Historically, the RFM (Recency, Frequency, Monetization) model has been employed with success in designing direct marketing campaigns Kohavi and Parekh (2004). While definitions vary, generally recency refers to when the last purchase was made. Frequency denotes number of purchases in a certain time period. Monetization can represent the amount of the last purchase, cumulative spending or the average amount spent per purchase.

The RFM model proposed by Hughes (1996) is often used: Customers are binned into 5 segments for each of the RFM features individually and labeled ordinal, resulting in 125 cells that can then be used to identify customers most likely to respond. The best customers have a high score for each of the 3 features. The drawback of this approach is that generally, marketing efforts go towards the best customer segment.

Over time, different extensions of RFM (Fader, Hardie, and Lee 2005; Yeh, Yang, and Ting 2009) and, increasingly, machine learning models such as Na”ive Bayes, Random Forests [Stubseid and Arandjelovic (2018); durango2013], Chi-squared automatic interaction detection and logistic regression were proposed. While in some situations, these alternatives outperformed RFA (McCarty and Hastak 2007), RFA remains popular because of its intuitive interpretation.

In this thesis, a radically data-driven approach was chosen. Several machine learning algorithms were employed and compared to predict potential donors and the net profit generated instead of building on previously developed, specialized models.

1.1. Task Background

A U.S. American veterans organization regularly conducts direct marketing campaigns, asking their members for donations (called gifts in the documentation) for the treatment of veterans with spinal injuries. The goal for the organization is to maximize net profit from their campaigns.

Only a small proportion of the members donate in reply to a campaign, while each letter sent out has a unit cost of 0.68 \$US. In order to maximize profit, it is therefore desirable to only mail members who are likely to donate.

1. Introduction

The members are grouped, among other criteria, by the recency of their last gift. Of these groups, the so-called *lapsed* donors are of particular interest. These are members who made their last gift to the organization 13 to 24 months prior to a given campaign. This group is important for two reasons: Firstly, the probability of a member donating decreases with the time the member has not donated. Enticing these lapsed donors to give again therefore maintains an active member base. Secondly, the organization has found that there is a negative correlation between the dollar amount donated and the probability to respond to a campaign. This means it is important to include the most unlikely donors in future mailings because if they donate, the net revenue is particularly large. If these unlikely donors would be suppressed from future campaigns, the gains from additional *small dollar* lapsed donors would not offset the losses from the potential *high dollar* donors.

The data at hand was distributed for the purpose of the KDD-CUP of the year 1998¹. The cup was until recently held yearly under the aegis of the special interest group on Knowledge Discovery and Data Mining (SIGKDD), which itself is part of the Association for Computing Machinery²(ACM).

1.2. Goals and Requirements

- The ultimate goal is to beat the winner of the original cup in terms of predicted net profit for the promotion. A complete data analysis including data preprocessing, model evaluation and -selection and final prediction has to be performed.
- Furthermore, the thesis should support future work on the data set by providing a solid basis especially on the preprocessing of the data.
- The supervisor of this thesis requested that the solution be demonstrated using Python as programming environment.

1.3. Conventions and Notes

- A member of the organization will be referred to as an *example*. Each example is described by a set of *features* (explanatory variables) and has two *targets* (dependent variables) associated.
- The *current* promotion refers to the most recent promotion made, *current* donors are those examples who donated in response to the promotion.
- A *lapsed* donor is an example who made their last donation between 13 and 24 months prior to the current promotion.
- Software packages are denoted as package with specific modules contained in packages written as `package.module.Class`. Where available, software used is cited with the article in which it was published. Less established packages are cited by giving their public source code repositories.
- All self-written code, including the reproducible analysis process, is published online at <https://github.com/datarian/master-thesis-msc-statistics>. These resources, especially the folder notebooks, form an integral part of this thesis.

¹For an archive of past cups, see SIGKDD - KDD Cup

²<https://acm.org>

2. Data

The data set contains data on all 191'779 members of the organization with a *lapsed* donation status (last donation 13 – 24 months ago) relative to the promotion sent out in June 1997 (named *current* promotion hereafter).

The data is provided¹ split in two sets, of which one is intended for learning (95'412 examples), the other for validation and final prediction (96'367 examples). The features are identical between the two except for the target features that have been separated from the validation set.

In this section, the *learning* data set will be characterized.

2.1. General Structure

The input data with $n = 95'412$ rows and $p = 479$ columns is structured as follows: $\mathbf{D} = \{\{\mathbf{x}_i, \mathbf{y}_i\}\}, i = 1 \dots n, \mathbf{x} \in \mathbb{R}^{p-3}, \mathbf{y} \in \mathbb{R}^2$.

There are $m = p - 3 = 476$ explanatory features, two targets and one unique identifier for each example.

The m features are grouped into four blocks of information:

- Member database with personal particulars, interests and organization-internal information on examples: 81 features
- Characteristics of example's neighborhood from the US census 1990: 286 features
- Promotion history: 54 features
 - Summary of promotions sent to an example in the 12 months prior to the current promotion
 - Sending dates and RFA status of promotions 13-36 months prior to current promotion
- Giving history: 57 features
 - Summary statistics
 - Responses to promotions 13-36 months prior to current promotion

2.2. Exploratory Data Analysis

Key insights from the exploratory data analysis are shown below. The detailed analysis can be studied online in the corresponding Jupyter notebook *3_EDA.ipynb*².

¹See <https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1998+Data>, accessed on 30.5.2019

²https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/3_EDA.ipynb

2. Data

2.2.1. Data Types

An analysis of the data set dictionary (Appendix B.2) reveals the following data types:

- Index: CONTROLN, unique record identifier
- Dates: 73 features in `yy-mm` format.
- Binary: 48 features
- Categorical: 46 features
- Numeric: 309

The data types present after import with the python package `pandas`' `read_csv()` are shown in Table 2.1. Two things are worth noting: There are many integer features, meaning that most numeric data is discrete. Few categorical features and no date features were automatically identified. The missed categoricals and dates are most likely in the group of `Object` features, which is `pandas`' catch-all type, and will have to be transformed during preprocessing.

Table 2.1.: Data types after import of raw csv data

	Data content	Number of features
Integer	Discrete features, no missing values	297
Float	Continuous features and discrete features with missing values	48
Categorical	Nominal and ordinal features	24
Object	Features with alphanumeric values	109
Total		478

2.2.2. Targets

Of the two targets, one is binary (TARGET_B), the other continuous (TARGET_D). The former is a binary response indicator for the current promotion. The latter represents the dollar amount donated in response to the current promotion and is 0.0 \$ in case of non-response.

As can be seen in Figure 2.1, the response rate is 5.1 %. This means the binary target is highly imbalanced. Extra care will have to be taken during model training to obtain a model with a low generalization error.

The distribution of the continuous target for $\text{TARGET}_D > 0.0$ \$ is shown in Figure 2.2. Evidently, most donations are smaller than 25 \$, the 50-percentile lying at 13 \$ and the mean at 15.62 \$. There are a few outliers for donations above 100 \$, making the distribution right-skewed. Being monetary amounts, the observed values are discrete rather than strictly continuous.

2. Data

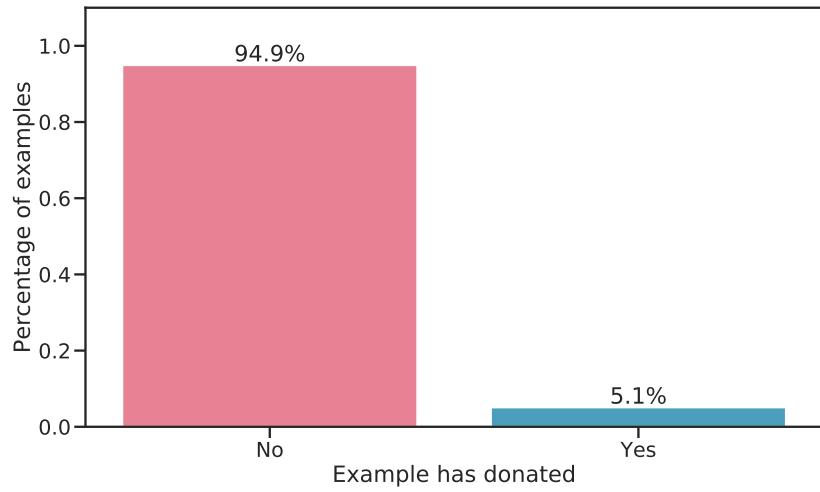


Figure 2.1.: Distribution of the binary target TARGET_B .

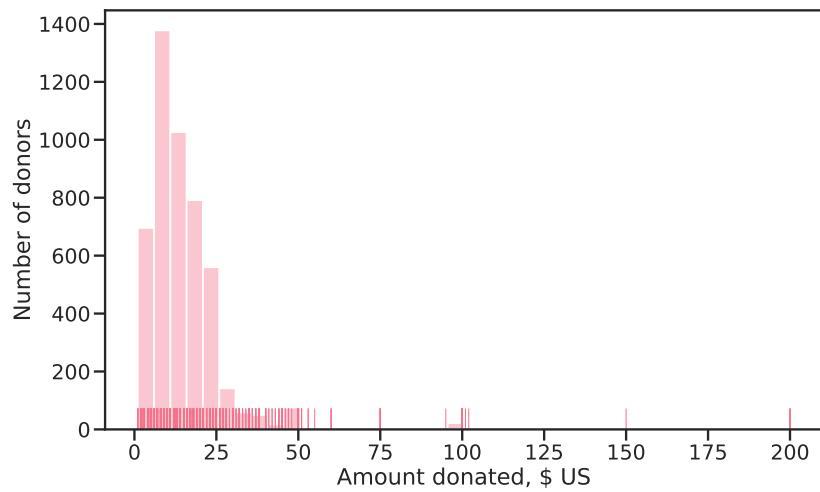


Figure 2.2.: Distribution of TARGET_D , the donation amount in \$ US (only amounts > 0.0 \$ are shown).

2. Data

2.2.3. Skewness

Most of the numerical features are skewed. Due to the high dimensionality, individual assessment of the features through boxplots or histograms was not feasible. Instead, skewness was measured with the Fisher-Pearson standardized moment coefficient $G_1 = \frac{\sqrt{n(n-1)}}{n-2} \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3 / s^3$ and plotted together with the $\alpha = 5\%$ confidence bound (CB) for a normal distribution (see Figure 2.3). Since G_1 is symmetric around zero, absolute values were chosen to display the results on a log scale. Evidently, no feature was found to be strictly normally distributed.

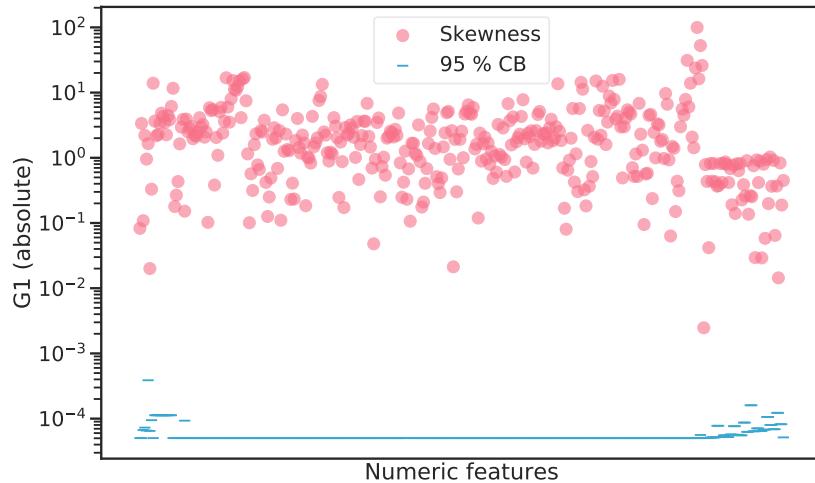


Figure 2.3.: Fisher-Pearson standardized moment coefficient (G_1) for all numeric features contained in the dataset. The confidence bound indicates the $\alpha = 5\%$ bound for the skewness of a normal distribution for any given feature.

The distributions of the 6 least skewed features (Figure 2.4) resemble normal (respectively binomial) or discrete uniform.

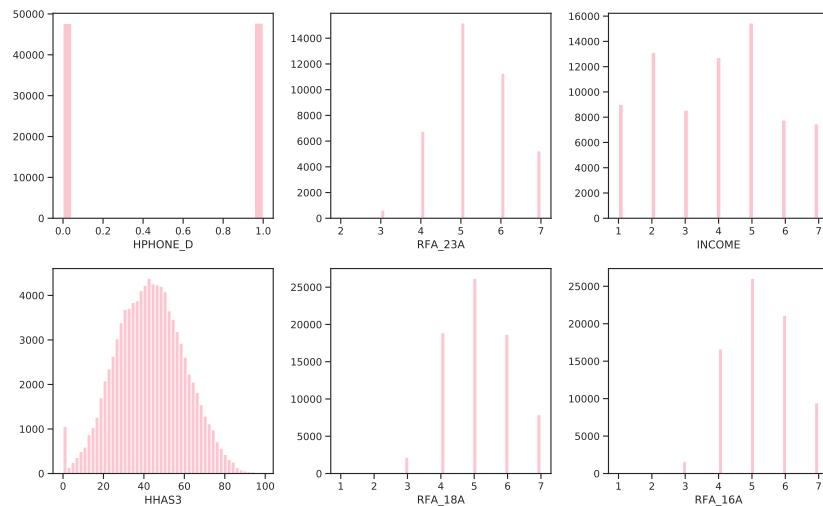


Figure 2.4.: Least skewed features by G_1 (adjusted Fisher-Pearson standardized moment coefficient).

2. Data

The 6 most skewed features (Figure 2.5) show heavily right-skewed poisson-like distributions which are the result of outliers.

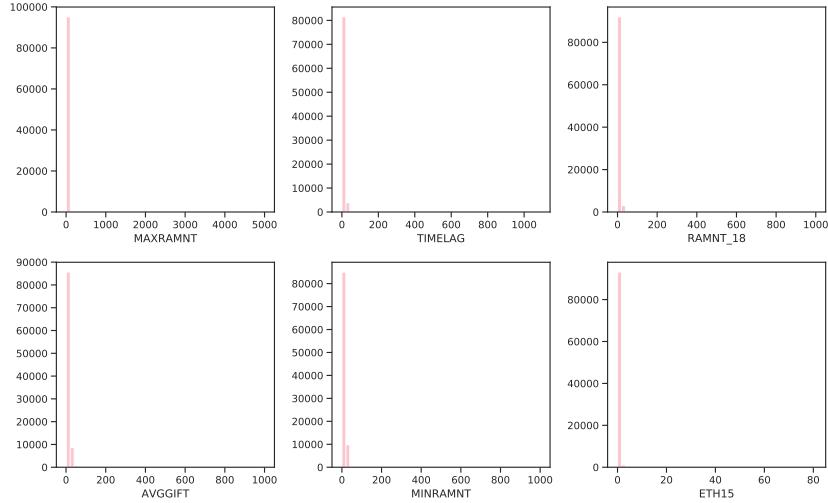


Figure 2.5.: Most skewed features by G1 (adjusted Fisher-Pearson standardized moment coefficient).

2.2.4. Correlations

The high dimensionality makes it hard to assess correlations in the data between individual features. A heatmap (see Figure 2.6) provides a high-level view. From left to right, three regions can be distinguished: First, there are member database features, followed by a large center region comprised of the U.S. census features, and rightmost, there are promotion and giving history features. Between these blocks, only few features are correlated. Within each block however, there is quite strongly correlated data.

2. Data

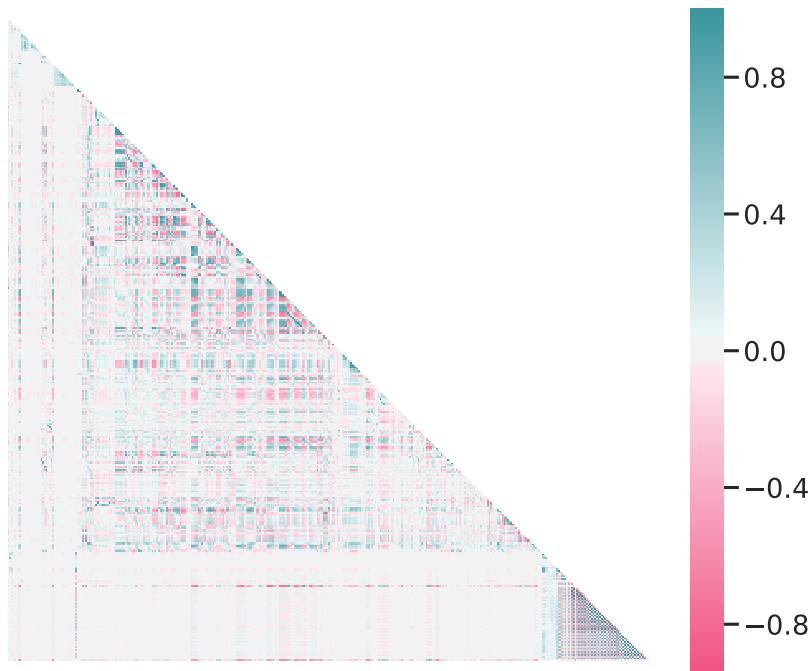


Figure 2.6.: Heatmap of feature correlations. Green means positive correlation, magenta means negative correlation. Perfect correlation occurs at 1.0 and -1.0.

2.2.5. Donation Patterns

When looking at the all-time recency-frequency-amount (RFA) features for donors (Figure 2.7), no clear trend is visible to discern current donors from non-donors. Current donors are found across the whole range of recency values, although slightly concentrated at the lower end (low recency is better). A slight correlation between frequency and current donation status is also discernible: Current donors tend to “float” on top of the “sedimented” non-donors. Those examples with the highest yearly donation amount did not donate in the current promotion.

The data set documentation states that donation amounts are positively correlated with recency, the time since the last donation. This means that the longer an example goes without donating, the higher the donation amount if it can be enticed into donating again. Figure (2.8) gives some evidence for this assumption. Starting from 15 months, the number of donations above 50 \$ increase.

There is another insight gained when considering the number of donations an example has made, indicated by the point size, and donation amount: Frequent donors give relatively small sums, while the largest donations come from examples who rarely donate.

There are many examples who donated within the last 12 months prior to the current promotion (Figures 2.7 and 2.8). This is a surprising observation. The data set should contain *lapsed* donors only, so there should be no donations recorded for that period. An explanation could be that the recency status is considered strictly for direct responses to promotions. Examples who donate regularly (i.e. monthly, yearly), irrespective of the promotions mailed out, would not be evaluated in terms of RFA under that assumption.

2. Data

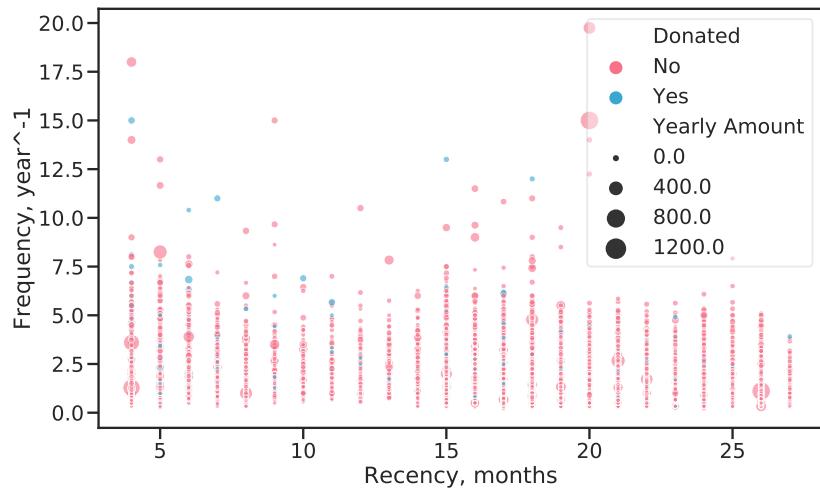


Figure 2.7.: Analysis of all-time RFA values by response to current promotion. *Recency* is the time in months since the last donation, *Frequency* the average number of donations per year and *Amount* the average yearly donation amount.

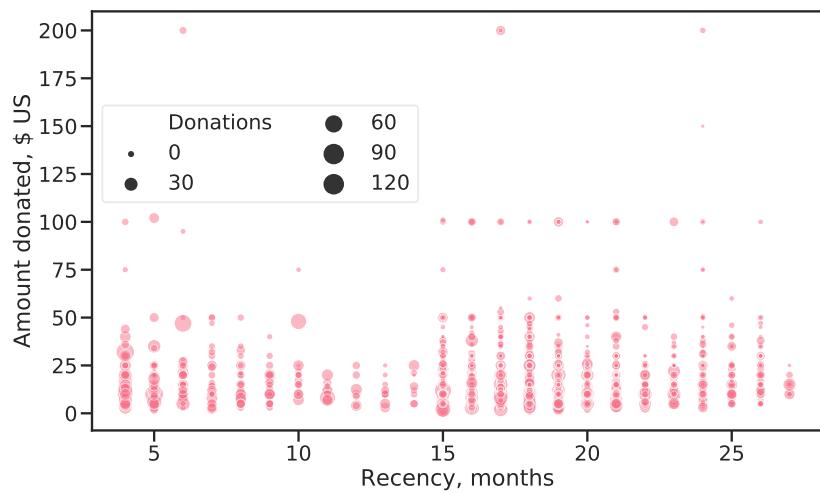


Figure 2.8.: Donation amount for the current promotion against months since last donation. The dot size indicates the number of times an example has donated.

2. Data

The data contains RFA features for each of the 24 promotions in the promotion history data, giving example's status per a given promotion. Considering the RFA features for the current promotion, the insights above can be supported. Since the data set contains only lapsed donors, the recency feature is constant and not of interest (it is a nominal feature, *lapsed* being one of the levels). Regarding the frequency of donations (number of donations 13 - 24 months prior to the promotion), shown in Figure 2.9, a clear trend is apparent. With increasing donation frequency, donation amounts decrease.

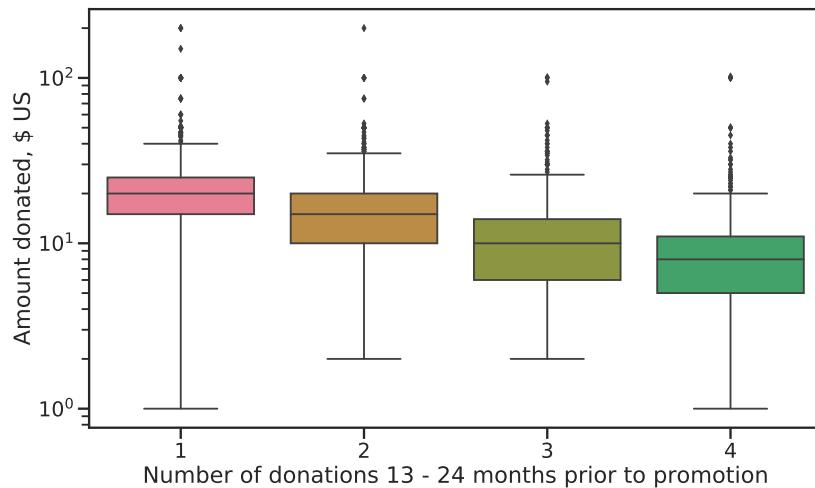


Figure 2.9.: Frequency of donations in the 13-24 months prior to current promotion against amount donated. Frequent donors give smaller amounts.

Figure 2.10 shows the geographical distribution of donations. The large urban centers like San Francisco, Los Angeles, Miami, Chicago and Detroit are clearly visible. To a lesser extent, cities like Houston, Dallas, Minneapolis, Atlanta, Tampa, Seattle and Phoenix can be made out. Examples living there give small amounts. Big donors (large total donations with a high average) can be made out in rural areas in the mid-west and Texas. Interestingly, only very few donations come from the north-eastern states.

Examples living in rural areas tend to donate larger sums, as Figure 2.11 shows. Shown are the living environments in progressively more rural settings against the average all-time donation amount per capita.

Socio-economic status by living environment reveals that examples with the highest status are rarely among the low-dollar donors. The median donation amount for the highest status is always higher than other status groups, but examples in the lowest status donate more than those in the medium level (Figure 2.12). The highest donations come from the medium status group.

2. Data

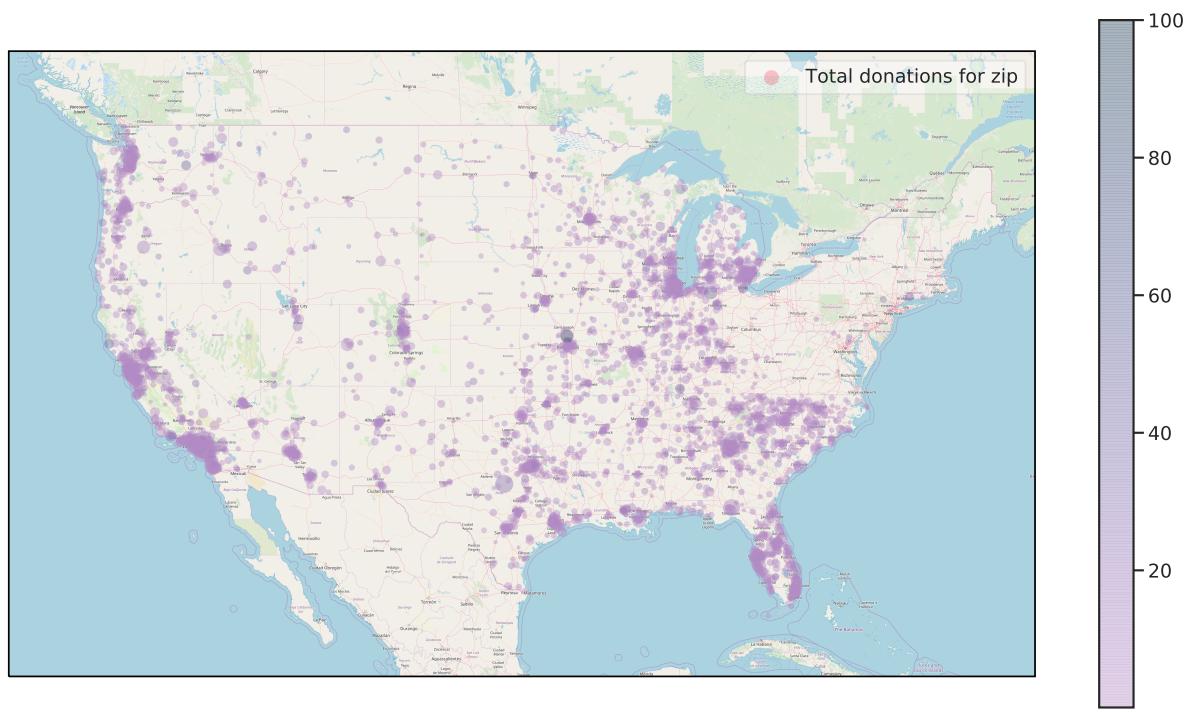


Figure 2.10.: Geographical distribution of donations by zip code. Point size indicates total donations for a zip code while the hue shows average donation amount.

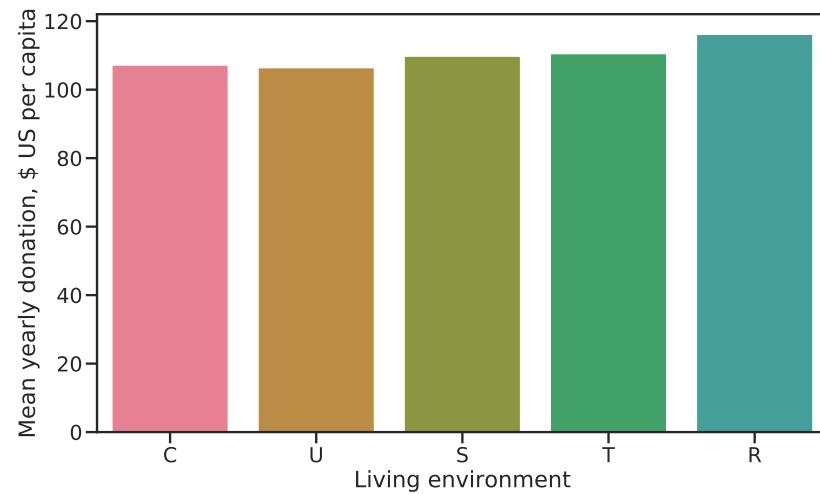


Figure 2.11.: Average cumulative donation amount per capita by living environment (C = city, U = urban, S = suburban, T = town, R = rural). The more rural, the higher the average donations.

2. Data

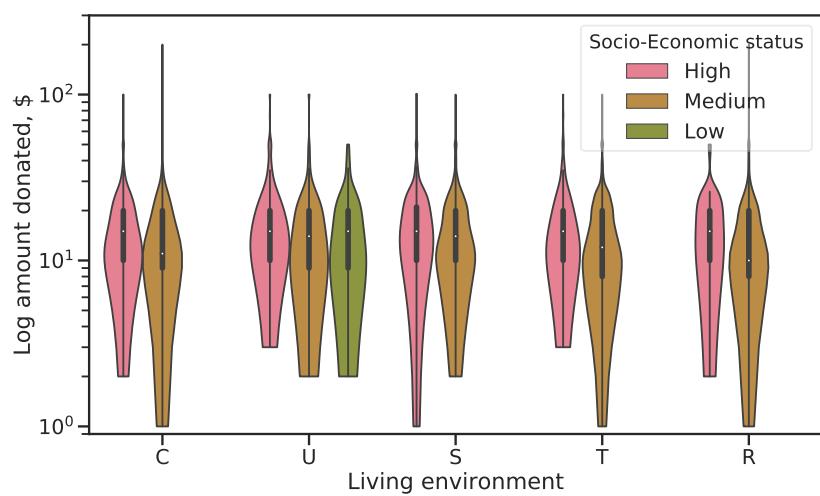


Figure 2.12.: Donation amount for current promotion by living environment and socio-economic status of examples. The violin plot shows the distribution of values similar to a kernel density estimation. Median values are indicated by white dots, the bold regions give the inner quartile range.

3. Experimental Setup and Methods

3.1. Tools Used

The problem itself was solved using the python language, making use of established packages. From the scipy ecosystem: `numpy`, a library for multidimensional arrays and vectorized operations on them (Oliphant 2006); `scipy`, a library for scientific computing (McKinney 2010); `pandas`, a library for data analysis (McKinney 2011); `matplotlib`, for high-quality plots (Hunter 2007). Furthermore: the machine learning library `scikit-learn` (Pedregosa et al. 2011) and the high-level data visualization library `seaborn` (Waskom et al. 2014).

Calculations were run in the cloud on a linux virtual machine¹.

Except for the development of a helper package, most programming was performed in interactive Jupyter notebooks (Kluyver et al. 2016).

The report was written in `rmarkdown` (Allaire et al. 2018) using `knitr` (Xie 2015) and `bookdown` (Xie 2016) to render the document into several output formats.

All work was tracked in version control.

3.2. Data Handling

The learning data set was used to establish the complete analysis pipeline, i.e. preprocessing, imputation and feature selection, model evaluation and -selection and prediction method. The test data was not touched until making the final prediction, subjecting it to the complete pipeline determined on the learning data.

In accordance with recommendations in Friedman, Hastie, and Tibshirani (2001), the learning data set was further split 80/20 into training and validation sets (see Figure 3.1). The training set was used to train different models while the validation set served to tune hyperparameters. The split was performed using a stratified sampling algorithm to preserve target class frequencies based on the response indicator.

¹For details, refer to: <https://docs.microsoft.com/en-us/azure/machine-learning/data-science-virtual-machine/overview>, accessed on 5.6.2019

3. Experimental Setup and Methods

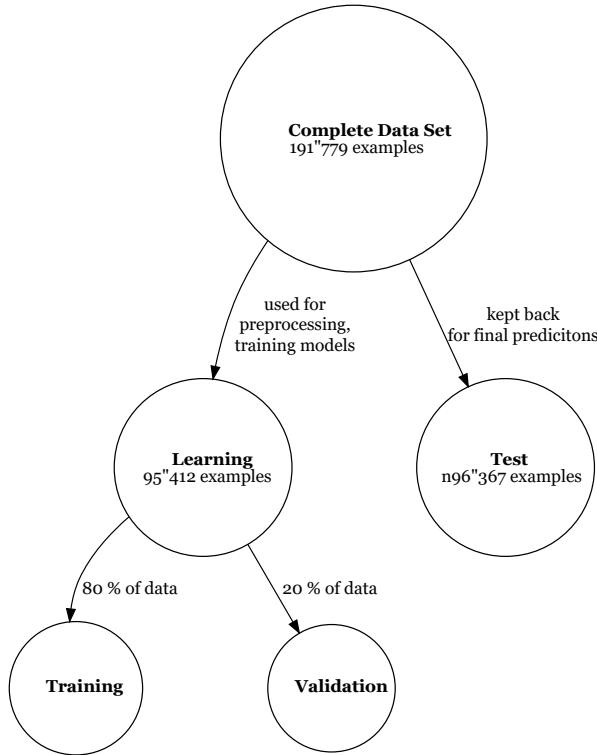


Figure 3.1.: Data set use for training and predictions.

3.3. Data Preprocessing

The necessary preprocessing was guided by practical necessity (input errors, inconsistent categories), the requirements of the algorithms that were examined (Section 3.5) and the requirements set out in the cup documentation:

- Only numeric features (required by some algorithms)
- Imputation of missing values (required by cup documentation)
- Removal of constant and *sparse* features (required by cup documentation)

The transformations were established interactively in Jupyter notebooks. Once finalized, transformations were implemented in the python package kdd98².

3.3.1. Cleaning

The transformations applied can be studied in the Jupyter notebook *1_Preprocessing.ipynb*³.

The cleaning stage of preprocessing encompassed the following transformations:

- Removing *noise*: Input errors, inconsistent encoding of binary / categorical features

²Available from <https://github.com/datarian/thesis-msc-statistics/tree/master/code/kdd98>

³https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/1_Preprocessing.ipynb

3. Experimental Setup and Methods

- Dropping constant and sparse (i.e. those where only few examples have a value set) features
- Imputation of values missing at random (MAR)

MAR values in the sense of Rubin (1976) are missing conditionally on other features in the data. For example, there are three related features from the promotion and giving history: *ADATE*, the date of mailing a promotion, *RDATE*, the date of receiving a donation in response to the promotion and *RAMOUNT*, the amount received. For missing *RAMOUNT* values, it can be checked if *RDATE* is non-missing. If *RDATE* is missing, then the example most likely has not donated and *RAMOUNT* can be set to zero. If, on the other hand, both date features have a value, *RAMOUNT* is truly missing.

3.3.2. Feature Engineering

The transformations applied during feature engineering are described in detail in the Jupyter notebook *2_Feature_Engineering.ipynb*⁴. The result of this step was an all-numeric data set usable for downstream learning.

All non-numeric (i.e. categorical) features were encoded as numeric representations. For ordinal features, manual mappings from alphanumeric levels to integer numbers were specified. For nominal features, two encoding techniques provided in package *categorical-encoding*⁵ were employed, depending on the number of levels:

- One-hot encoding for ≤ 10 levels: For each level of a categorical feature, a new feature is created. An additional feature may be added to indicate missing values. Exactly one of these new features is set to 1, indicating the original level.
- Binary encoding, > 10 levels: The levels of the categorical are first transformed to ordinal (i.e. to a sequence of integer numbers). Then, these numbers are taken to the base of 2. (A 5, for example, becomes 101). According to the number of levels, new features for the binary digits are created. As an example: To represent 60 levels, 6 features are required ($2^6 = 64$).

Also, several features were transformed to better usable formats (i.e. converting dates to time deltas, converting zip codes to coordinates). Care was taken to keep the dimensionality of the data set as low as possible.

3.3.3. Imputation

Three different approaches were evaluated. The details are shown in Jupyter notebook *4_Imputation.ipynb*⁶.

⁴https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/2_Feature_Engineering.ipynb

⁵Available at: <https://github.com/scikit-learn-contrib/categorical-encoding/>, accessed on 5.6.2019

⁶https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/4_Imputation.ipynb

3. Experimental Setup and Methods

3.3.3.1. K-Nearest Neighbors

kNN imputation uses those examples that are “near” an example with a missing value in feature f to impute the value. In short, the kNN algorithm by Troyanskaya et al. (2001) implemented in `missingpy`⁷ works as follows:

1. Construct the distance matrix D with distances between examples.
2. Order all features descending by number of missing values.
3. Starting with the feature with most missing, for each example with a missing value, use the k nearest neighbors’ mean or median to impute.

The algorithm runs until all values are imputed.

While very attractive because of the intuitive approach and because it preserves data characteristics (continuous / discrete, categoricals) in the features, the distance matrix is very memory-intensive for large data sets. Also, it is required to remove features with $> 80\%$ missing values first.

3.3.3.2. Iterative imputation

Iterative imputation, implemented in package `fancyimpute`⁸, works similar to the R-package `mice` (see van Buuren and Groothuis-Oudshoorn (2011)). Before imputation, all features have to be transformed to numerical data types and normalized because the underlying model is a linear regression.

In short, the algorithm works as follows:

1. Features are ordered by the fraction of missing values
2. Starting with the feature with most missing values, use the other features to build a linear model, using the current feature as the dependent variable and predict missing values.
3. Repeat step 2 until all features are complete
4. Repeat steps 2 – 3 n times, $n = 5$ was chosen

3.3.3.3. Simple Imputation and Categorical Indicator

This approach is straightforward: Numeric features are imputed by their median value to make the imputation robust to skewed distributions.

As the algorithm used (`sklearn.impute.SimpleImputer`) only supports numerical data types, categorical features were treated separately during feature engineering (Section 3.3.2): The one-hot or binary encoded categoricals had one more feature added, indicating missing values.

⁷Available at: <https://github.com/epsilon-machine/missingpy>, accessed on 11.6.2019

⁸Available at: <https://pypi.org/project/fancyimpute/>, accessed on 30.06.2019

3. Experimental Setup and Methods

3.3.4. Feature Selection

One of the biggest caveats in dealing with high-dimensional data is the infamous “Curse of Dimensionality” coined by Bellman (1966). The curse comes from the fact that with an increasing number of dimensions of the feature space, the number of possible combinations grows exponentially. In order to cover all possible combinations with several examples, a huge amount of data would be required as a result. In the area of machine learning, high dimensionality frequently manifests in the form of overfitting, which leads to an unacceptably large generalization error Goodfellow, Bengio, and Courville (2016). Hughes (1968) showed that for a fixed number of examples, model performance first increases with increasing number of dimensions but then decreases again.

Boruta, introduced by Kursa, Rudnicki, and others (2010) in the form of an R package, performs feature selection by solving the so-called all-relevant feature problem. The algorithm was found to perform very well regarding selection of relevant features in Kursa and Rudnicki (2011).

In short, the algorithm works as follows:

1. The input matrix \mathbf{X} of dimension $n \times p$ is extended with p so-called *shadow features*. The shadow features are permuted copies of the features in \mathbf{X} . They are therefore de-correlated with the target.
2. On the resulting matrix \mathbf{X}^* , a random forest classifier is trained and the Z-scores $\frac{\text{loss}}{sd}$ for each of the $2p$ features calculated.
3. The highest Z-score among the shadow features $MZSA$ is determined.
4. All original features are compared against $MZSA$ and those features with a higher score selected as important.
5. With the remaining features, a two-sided test for equality of the Z-scores with $MZSA$ is performed and all features with significantly lower score are deemed unimportant.
6. All shadow copies are removed, go to step 1.

The algorithm terminates when all attributes are marked as either important or not important or when the maximum number of iterations is reached.

For this thesis, a python implementation⁹ was used.

The models were all learned on the same data set resulting from the boruta selection.

3.4. Prediction

The desired quantity to estimate is net profit $\hat{\Pi}$. In order to predict this quantity, a two-step procedure was implemented, utilizing the response indicator $TARGET_B$ and the continuous target $TARGET_D$, respectively. For each step, a model was trained. One is a classifier, predicting \hat{y}_b , the probability of donating. The other is a regressor, predicting the donation amount \hat{y}_d . The classifier was trained on the complete learning data set, while the regressor was trained on $\mathbf{X}_d = \{x_i | y_{b,i} = 1, i = 1 \dots n\}$ and $y_d = \{y_{d,i} | y_{b,i} = 1, i = 1 \dots n\}$. The non-random sample leads to a conditionally predicted donation amount, thereby introducing bias. A correction for the bias is necessary as a consequence. This approach resembles Heckman (1976)'s two-stage procedure which is widely used in econometrics. Heckman's procedure was presented for the question of wage offerings for women. Data on wages was

⁹Available from https://github.com/scikit-learn-contrib/boruta_py, accessed on 5.6.2019

3. Experimental Setup and Methods

only available for working women, thereby introducing bias if wage offerings were only predicted on this sample. The question of whether an example works or not is seen as an unobserved feature. In Heckman (1976), a probit model is used for this first *selection* stage. The inverse Mills ratio of the probit, $\frac{\phi(\hat{y}_i - \mathbf{X}\beta)}{\Phi(\hat{y}_i - \mathbf{X}\beta)}$, is calculated and included in the data for predictions in the second *observation* stage with OLS regression.

3.4.1. Setup of the Two-Stage Prediction

For the first stage some classifier is used, predicting the probability for example x_i to donate. This is not a probit ($P(Y = 1|X) = \Phi(X^T\beta)$), of course. Instead, the resulting distribution depends on the classifier.

$$\hat{y}_b = f(\mathbf{X}) \quad (3.1)$$

where \hat{y}_b is the vector of predicted probabilities of donating and f is the classifier.

The second stage is performed on \mathbf{X}_d and consists in predicting the donation amount using a regression model:

$$\hat{y}_{dt} = g(\mathbf{X}_d) \quad (3.2)$$

where $\hat{\mathbf{y}}_d$ is the vector of predicted donation amounts and g is the conditionally learned regression model. g is learned with Box-Cox transformed target \mathbf{y}_d , so \hat{y}_{dt} is also Box-Cox transformed with parameter λ .

The decision of whether to include example i in the promotion is governed by the following indicator function. In it, α^* accounts for the introduced bias. Every example that has a predicted donation amount of more than the unit cost is included.

$$\mathbb{1}_{\hat{y}_{i,b} * \exp(\hat{y}_{i,dt}) * \alpha^* > \exp(u_t)}(\hat{y}_{i,dt}) \quad (3.3)$$

where $\alpha^* \in [0, 1]$ is a factor to correct for bias introduced due to the non-randomness of \mathbf{X}_d , $\hat{y}_{i,dt}$ is the predicted donation amount, transformed so as to normalize the distribution learned beforehand and u_t is the unit cost, Box-Cox transformed with parameter λ . The exponential is used to deal with negative values resulting from the Box-Cox transformation.

Finally, the quantity estimated is net profit $\hat{\Pi}$. It is calculated by summing over the product of the indicator function (3.3) and the net profit for examples 1 ... n . For unseen data, @eq:pi-alpha is altered by using the estimated net profit $\hat{y}_{d,i}$ in the product.

$$\hat{\Pi}_\alpha = \sum_{i=1}^n \mathbb{1}_{\hat{y}_{i,b} * \exp(\hat{y}_{i,dt}) * \alpha^* > \exp(u_t)}(\hat{y}_{i,dt}) * (y_{i,d} - u) \quad (3.4)$$

3. Experimental Setup and Methods

3.4.2. Optimization of α^*

With equation (3.4), the estimated profit Π was calculated on the learning data for a grid of α values, $\alpha \in [0, 1]$. The optimal value was then $\alpha^* = \underset{\alpha}{\operatorname{argmax}} f(\alpha)$ where f is a function that was fit to Π .

For f , a cubic spline s was used. α^* was determined as follows:

1. Fit $s(\Pi)$, the cubic spline on the estimated profits for the grid of α values
2. Derive $ds = \frac{\delta}{\delta \alpha} s$
3. Find the finite roots of ds , $\alpha_{\text{candidates}}$, representing candidates for α^*
4. Determine $\alpha^* = \underset{\alpha}{\operatorname{argmax}} s(\alpha_{\text{candidates}})$

3.5. Model Evaluation and -Selection

Several algorithms (estimators) were trained in parallel. Once good hyperparameters were found using randomized grid search, performance of the estimators was compared using a common metric in order to select the best (see Figure 3.2 for a schematic process overview). This was done independently for classifiers (predicting the binary target) and regressors (predicting the continuous target). The process is documented in notebook `6_Model_Evaluation_Selection.ipynb`¹⁰.

The pipeline functionality of `scikit-learn` was used to combine preliminary data transformations, i.e. scaling of features (where necessary) and resampling with the estimator-algorithm. This allows to jointly tune hyperparameters for transformations and the estimator.

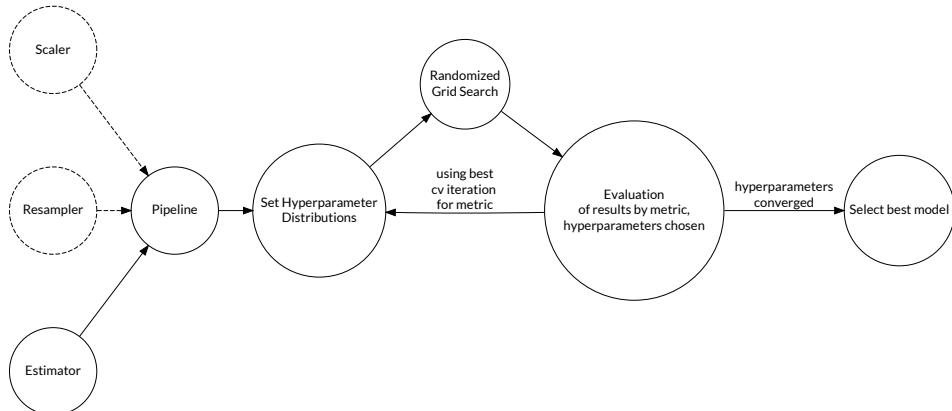


Figure 3.2.: Learning process schematic.

A common random seed was used for all algorithms and random number generators.

¹⁰https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/6_Model_Evaluation_Selection.ipynb

3. Experimental Setup and Methods

3.5.1. Evaluation

Randomized grid search was run with 10-fold cross-validation (CV). The best-performing pipeline for each algorithm was stored in a python dictionary during evaluation. The dictionary was persisted to disk and only updated when an algorithm's metric improved. This ensured that the best hyperparameter settings were always retained during the extensive model evaluation phase.

3.5.1.1. Randomized Grid Search

In randomized grid search, (Bergstra and Bengio 2012), probability distributions for hyperparameter values are specified. The algorithm then runs a defined number (10 were used) of random combinations by sampling from the distributions. Compared to the usual grid search, this can greatly speed up the learning process because good hyperparameter settings are generally identified with less iterations.

After one round of learning, the hyperparameter distributions were adjusted before the next iteration as follows: When the best value was found near the limits of the domain, the distribution was shifted in this direction. For values falling inside the domain, the distribution was narrowed down towards the found value. This procedure was repeated until the hyperparameters converged.

The implementation of randomized grid search in `scikit-learn` returns a summary table with the CV results for each hyperparameter combination. By comparing the mean test score for the chosen metric, the best hyperparameters can be determined.

3.5.1.2. Cross-Validation

CV splits the training data into several *folds* of equal size. The algorithm is trained as many times as there are folds, holding back one of the folds at each training step for validation using some specified performance metric and training with the rest of the data. This procedure enables quantification of the generalization error and the calculation of statistics that indicate the variance of the model. Following guidance in Kohavi and others (1995), 10-fold CV was used, which trades higher bias for lower variance compared to fewer folds.

3.5.2. Selection

3.5.2.1. Classifiers

Both recall and F1 were initially selected and calculated during model evaluation. In the end, recall was used to select the best classifier as it produced better estimators for the task. Reasoning for the choice of these metrics is given below.

For classification problems, the confusion matrix (see Figure 3.3) can be used to construct various performance metrics. A true positive (TP) indicates a correctly predicted 1, a false negative (FN) is a falsely predicted 0, a false positive is a falsely predicted 1 and finally a true negative (TN) is a correctly predicted 0.

For the data analyzed here, *1* means an example has donated, *0* means the example has not donated.

3. Experimental Setup and Methods

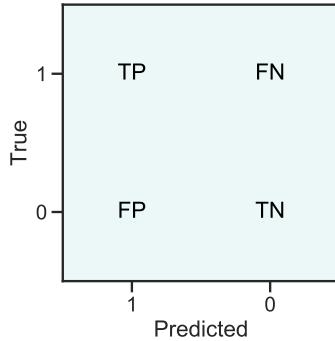


Figure 3.3.: Definition of the confusion matrix for a two-class problem.

The definitions of some often-used metrics are given below¹¹. The choice of metric depends on the goal of the prediction and the data at hand.

$$\begin{aligned}
 \text{Recall / Sensitivity / True Positive Rate TPR} &= \frac{TP}{TP + FN} \\
 \text{Specificity / True Negative Rate TNR} &= \frac{TN}{TN + FP} \\
 \text{Precision / Positive Predictive Value PPV} &= \frac{TP}{TP + FP} \\
 \text{Negative Predictive Value NPV} &= \frac{TN}{TN + FN} \\
 \text{False Negative Rate FNR} &= \frac{FN}{FN + TP} \\
 \text{False Positive Rate FPR} &= \frac{FP}{FP + TN} \\
 \text{Accuracy} &= \frac{TP + TN}{TP + FP + FN + TN} \\
 \text{F1 score} &= \frac{2TP}{2TP + FP + FN}
 \end{aligned}$$

The goal, as mentioned earlier, is to maximize net profit. To achieve this, a balance between predicting as many TP's as possible while keeping the number of FP's low has to be found. One FP costs 0.68 \\$ (sending a letter to a non-donor). Keeping in mind the distribution of TARGET_D (Section 2.2.2), one FN means loosing at least 0.32 \\$ of possible profit (not sending a promotion to a donor, smallest donation amount is 1 \\$). The expected loss in profit for one FN is even 15 \\$ (corresponding to the mean donation amount), which means that with each TP, the cost of 22 FP's can be covered on average.

The default metric for classification is accuracy, but in the case of imbalanced targets, it is not a desirable metric (TN is present both in the nominator and denominator, dominating the score). The metrics that could be used beneficially because they involve TP an not TN are F1, recall and precision. Since in precision, the FP of the majority class have the highest influence from among the three, it was discarded from the candidate list.

¹¹taken from https://en.wikipedia.org/wiki/Confusion_matrix, accessed on 28.05.2019

3. Experimental Setup and Methods

3.5.2.2. Regressors

For regression, $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$ was used, mainly because it is the default metric for regression algorithms in `scikit-learn`. R^2 has the drawback of depending on the variance of the data used to fit the model and therefore is different for other data. It was however assumed that because learning and test data have the same generating function, R^2 can be used to select a regression model.

3.5.3. Dealing With Imbalanced Data

Several different approaches were explored. The following over-/undersampling techniques available in package `imblearn` by Lemaître, Nogueira, and Aridas (2017) were studied:

- Random oversampling of the minority class
- Random undersampling of the majority class
- SMOTE (synthetic minority oversampling technique), variant borderline-1

The random sampling algorithms either draw from the minority class with repetition or draw random samples from the majority class without repetition until the labels are balanced. SMOTE does not sample from the data but instead generates synthetic samples from the minority class, thereby countering the danger of overfitting by learning on a small number of repeated observations (random oversampling) or a small subset of the training data available (random undersampling). The SMOTE variant *borderline-1* chosen generates samples that are close to the optimal decision boundary, where misclassification is likely.

Additional experiments were run with class- and sample-weights set on the data without resampling. For class weights, the ratio of non-donors vs. donors was used. For sample weights, the donation amount was employed, rescaled to the interval $[0, 1]$.

3.5.4. Algorithms

A short introduction of each algorithm is given below. For each algorithm, the hyperparameters that were tuned during learning are given. The choice of algorithms was made so as to cover a wide range of underlying concepts.

3.5.4.1. Random Forest

Random forest (RF) belongs to the family of so-called ensemble learners and was introduced by Breiman (2001). Predictions are made by majority vote of an ensemble of decision trees (CART, Breiman et al. (1984)). The RF can be employed both for regression and classification tasks. RF's are insensitive towards scale differences in the individual features, and, depending on the implementation, can deal with missing values. Another important feature of RF is the assessment of *variable importance* by summing the loss improvement for each split in every tree per feature (Friedman, Hastie, and Tibshirani 2001).

3. Experimental Setup and Methods

During learning, a random sample of the available features is drawn with replacement for each tree (bagging, or bootstrap aggregating, see Breiman (1996)), thereby reducing the variance of the ensemble estimator. Furthermore, splits within a tree are determined again on a random subset of the features. These sources of randomness tend to increase bias of the forest, yet the decrease in variance due to averaging through majority vote outweighs the bias increase Friedman, Hastie, and Tibshirani (2001). Breiman (2001) shows that as the forest grows, the generalization error converges almost surely. This means that random forests are insensitive to overfitting and perform better the more trees are grown.

As explained in Friedman, Hastie, and Tibshirani (2001), trees are grown as follows:

For data with n examples and p features $D = \{\{x_i, y_i\}, i = 1 \dots n, x_i = \{x_{i,1}, x_{i,2}, \dots x_{i,p}\}\}$, the CART algorithm decides on the structure of the tree, the splitting features and the split points. As a result, the data is partitioned into M regions R_1, R_2, \dots, R_M .

Regression The response \hat{y} is modeled as a constant c_m for each region:

$$f(x) = \sum_{m=1}^M c_m \mathbb{1}(x \in R_m) \quad (3.5)$$

Using as loss criterion the sum of squares $\sum_{i=1}^n (y_i - f(x_i))^2$, the best \hat{c}_m is the average of y_i in the region:

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m). \quad (3.6)$$

The algorithm greedily decides on the best partition. Starting with all data, a splitting feature j and a split point s is considered, creating two regions $R_1(j, s) = \{X | X_j \leq s\}, R_2(j, s) = \{X | X_j > s\}$. Feature j and split point s are chosen by solving

$$\min_{j,s} \left[\min_{c1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \quad (3.7)$$

with the inner minimization solved using (3.6).

Classification For a binary classification problem with outcomes $\{0, 1\}$, predictions are made through the proportion of the positive class in a region R_m with N_m examples x_i inside, which is given by:

$$\hat{p}_m = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}(y_i = 1) \quad (3.8)$$

When $\hat{p}_m > 0.5$, the positive class is chosen.

The loss is described by impurity. When making a split, the feature j resulting in the highest impurity decrease is selected. Impurity is measured by the *Gini index*. For binary classification: A node m , representing region R_m with N_m observations has as proportion of the positive class $\hat{p}_m = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}(y_i = 1)$. The Gini index is then defined as $2p(1 - p)$. So the decision function is:

$$\min_{j,s} (\min_{Gini_l} \hat{p}_l + \min_{Gini_r} \hat{p}_r) \quad (3.9)$$

for regions R_l, R_r below the node.

3. Experimental Setup and Methods

The `RandomForestClassifier` and `RandomForestRegressor` included in `scikit-learn` were used for learning.

Hyperparameters

- `max_depth`, $\{1, 2, 3, \dots\}$: depth of the trees, 2^n leafs maximum. Controls tree size.
- `min_samples_split`, $\{2, 3, 4, \dots\}$: Minimum number of samples required to split a node, controls tree size.
- `max_features`, $\{1, 2, \dots, m\}$: Maximum number of the m features to consider when searching for a split. Friedman, Hastie, and Tibshirani (2001) recommend values in $m = \{1, 2, \dots, \sqrt{m}\}$, but for high dimensional data with few relevant features, larger m can lead to better results because the probability of including relevant features increases.
- `n_estimators`, $\{1, 2, \dots\}$: Number of trees to grow. In combination with early stopping, this can be set to a high value since learning will stop when the loss converges.
- `class_weight` {balanced, 1, 2, ...}: Weights on target classes: “balanced” calculates weights according to class frequencies, integer values specify weight on majority class relative to minority

3.5.4.2. Gradient Boosting Machine

The main idea behind boosting is to sequentially train an ensemble of weak learners which on their own are only slightly better than a random decision. The predictions of the individual weak learners are then combined into a majority vote (Kearns 1988).

Gradient boosting machine (GBM) extends on this idea. Like a random forest, GBM learns an ensemble of decision trees. However, trees are learned in an additive manner. At each iteration, the tree that improves the model most (i.e. in the direction of the gradient of the loss function) is added. For this thesis, the package `XGBoost` by Chen and Guestrin (2016) was used. They describe the algorithm as follows:

Assume a data set with n examples and p features: $D = \{\{x_i, y_i\}, i = 1 \dots n, x_i = \{x_{i,1}, x_{i,2}, \dots x_{i,p}\}\}$. The implementation uses a tree ensemble with K regression trees to predict the outcome for an example in the data by summing up the weights predicted by each tree:

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (3.10)$$

where $F = \{f(x) = w_{q(x)}\} (q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ is the space of regression trees. T is the number of leaves in a tree, q is the structure of each tree, mapping an example to the corresponding leaf index. Each tree f_k has an independent structure q and weights w at the terminal leafs.

For growing the trees in F , the following loss function is minimized:

$$L(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (3.11)$$

3. Experimental Setup and Methods

Here, l is a differentiable, convex loss function that measures the difference between predictions and true values. Since l is convex, a global minimum is guaranteed. $\Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$ is a penalty on the complexity of the trees to counter over-fitting. The algorithm thus features integrated regularization.

Now, at each iteration t , the tree f_t that improves the model most is added. For this, $f_t(\mathbf{x}_i)$ is added to the predictions at $t - 1$ and compute the loss:

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{t-1} + f_t(\mathbf{x}_i)) + \Omega(f_t) \quad (3.12)$$

To find the best f_t to add, the gradients finally come into play. With g_i and h_i the first- and second-order gradient statistics of l , the loss function becomes:

$$\tilde{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{t-1} + g_i f_t(x_i) + h_i f_t^2(x_i)) + \Omega(f_t) \quad (3.13)$$

Hyperparameters

- `learning_rate`, [0, 1]: *Shrinkage*, decreases step size for the gradient descent when $\eta < 1.0$, helping convergence. The number of estimators f_k has to be increased for small learning rates in order for the algorithm to converge.
- `min_child_weight`, [0, inf): Minimum sum of weights of the hessian in a node. When close to zero, the node is pure. Controls regularization.
- `subsample`, [0, 1]: A random sample from the n examples of size $s, s < n$ is drawn for each iteration, countering overfitting and speeding up learning.
- `colsample_by_tree`, [0, 1]: A random sample of the m features is drawn for growing each tree.
- `n_iter_no_change`, {0, 1, 2, 3, ...}: Early stopping. Based on an evaluation set, learning stops when no improvement on the performance metric (misclassification error was chosen) is made for a fixed number of steps.

3.5.4.3. GLMnet

The GLMnet is an implementation of a generalized linear model (GLM) with penalized maximum likelihood by Hastie and Qian (2014). Regularization is achieved through L^2 (ridge) and L^1 (lasso) penalties or their combination known as elastic net (Zou and Hastie 2005).

The loss functions are described in Hastie and Qian (2014). For the binary classification task at hand, a logistic regression was performed. The logistic regression model for a two-class response $G = \{0, 1\}$ with target $y_i = \mathbb{1}(g_i = 1)$ is:

$$P(G = 1|X = x) = \frac{e^{\beta_0 + \beta^T x}}{1 + e^{\beta_0 + \beta^T x}} \text{ or, in the log-odds transformation: } \log \frac{P(G=1|X=x)}{P(G=0|X=x)} = \beta_0 + \beta^T x.$$

The GLMnet loss function for logistic regression:

$$\min_{\beta_0, \beta} \frac{1}{N} \sum_{i=1}^N w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda [(1 - \alpha)||\beta||_2^2/2 + \alpha||\beta||_1] \quad (3.14)$$

3. Experimental Setup and Methods

where w_i are individual sample weights, $l(\cdot)$ the (negative) log-likelihood of the parameters β given the data, λ the amount of penalization and $\alpha \in [0, 1]$ the elastic net parameter, for $\alpha = 0$ pure ridge and for $\alpha = 1$ pure lasso.

For the regression task, a gaussian family model was used, having loss function:

$$\min_{(\beta_0, \beta) \in \mathbb{R}^{p+1}} \frac{1}{2N} \sum_{i=1}^N (y_i - \beta_0 - x_i^T \beta)^2 + \lambda [(1-\alpha)\|\beta\|_2^2/2 + \alpha\|\beta\|_1] \quad (3.15)$$

where $l()$ the (negative) log-likelihood of the parameters β given the data, λ the amount of penalization and $\alpha \in [0, 1]$ the elastic net parameter, for $\alpha = 0$ pure ridge and for $\alpha = 1$ pure lasso.

For learning, GLMnet evaluates many different λ values for a given α through cross validation. Because GLMnet is sensitive to scale differences in the features, input data (features and target) should be transformed to mean zero and unit variance.

Hyperparameters

- `n_splits`, {3, 4, 5, ...}: Number of CV-splits. Typical values are 3, 5 and 10.
- α , [0, 1]: parametrizes the elastic net. For $\alpha = 0$ pure ridge, for $\alpha = 1$ pure lasso.
- `scoring`: Scoring method for cross-validation (log-loss, classification error, accuracy, precision, recall, average precision, roc-auc)

3.5.4.4. Multilayer Perceptron

The multilayer perceptron (MLP) is a so-called feed-forward neural network. The network consists of at least three layers: an input layer, an arbitrary number of hidden layers and an output layer. Each layer is made up of units. The term feed-forward means that information flows from the input layer through intermediary steps and then to the output. The goal is to approximate the function f^* . For a classifier, $y = f^*(x)$ maps an example x to a category y . A feed-forward network defines a mapping $y = f(x, w)$ and learns the weights w by approximating the function f (Goodfellow, Bengio, and Courville 2016).

For a binary classification problem on a dataset with n examples and p features $D = \{\{x_i, y_i\}\}, x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,p}\}, y_i \in \{0, 1\}, i = 1 \dots n\}$, the input layer has p units, the output layer has 1 unit. The hidden layers each have an arbitrary number of *hidden* units.

Each unit, except for the input layer, consists of a perceptron, which is in effect a linear model with some non-linear activation function applied:

$$y = \phi(w^T x + b) \quad (3.16)$$

where ϕ is a non-linear activation function, w is the vector of weights, x is the vector of inputs and b is the bias. For ϕ , typical functions are the hyperbolic tangens $tanh(\cdot)$, the logistic sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$, or, more recently, the rectified linear unit $relu(x) = max(0, x)$ (Hahnloser et al. 2000; Goodfellow, Bengio, and Courville 2016).

During learning, the training examples are fed to the network sequentially. For each example, the prediction error is calculated using a loss function, which is typically the negative log-likelihood.

3. Experimental Setup and Methods

Then, the partial derivatives of the loss function with respect to the weights are computed for each unit and the parameters updated using stochastic gradient descent. This process is called back-propagation.

The complete network is a chain of functions, for the network trained here, it is:

$$\mathbf{y} = \phi^{(3)}(\mathbf{W}^{(3)T} \square^{(2)}(\mathbf{W}^{(2)T} \square^{(1)}(\mathbf{W}^{(1)T} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}) + \mathbf{b}^{(3)}) \quad (3.17)$$

where x is the vector of input features, y is the vector of outputs, $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}$ are the weight matrices for each layer and $b^{(1)}, b^{(2)}, b^{(3)}$ are the bias vectors for each layer and $\phi^{(1)}, \phi^{(2)}, \phi^{(3)}$ are the sets of perceptrons in the corresponding layer.

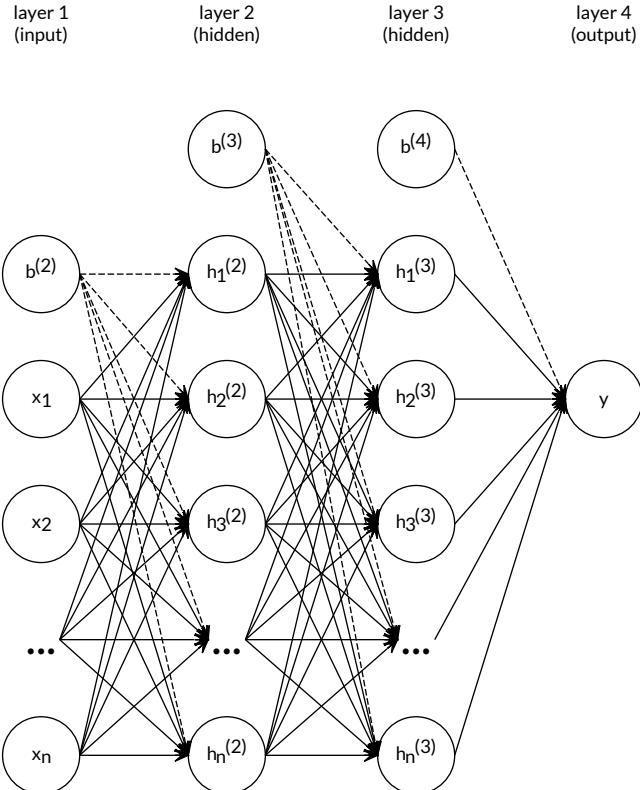


Figure 3.4.: Neural network topology used. Two hidden layers $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}$ are contained. $\mathbf{b}^{(2)}, \mathbf{b}^{(3)}$ and $\mathbf{b}^{(4)}$ are the bias vectors for the respective layers.

For this thesis, the scikit-learn implementation `sklearn.neural_network.MLPClassifier` was used with default settings (rectified linear unit as activation function, `adam` (stochastic gradient descent) solver). The network topology was determined by treating it as a hyperparameter. Best results were achieved with a network featuring two hidden layers with 28 hidden units each. The network is shown symbolically in Figure 3.4.

Hyperparameters

- `hidden_layer_sizes`: Tuples, i.e. (100,50,) for a network with 2 hidden layers of 100 and 50 units and one output unit.
- $\alpha, [0,1]$: L^2 regularization amount

3. Experimental Setup and Methods

- `learning_rate_init > 0`: Controls step size for the solver during back-propagation (regularization).

3.5.4.5. Support Vector Machine

In the case of binary classification, support vector classifiers (SVC) find a separating hyperplane $\{x : f(x) = x^T \beta + \beta_0 = 0\}$ with classification rule $G(x) = \text{sign}(x^T \beta + \beta_0)$. The best hyperplane is such that a margin M defined by parallel hyperplanes on either side is maximized. The larger the margin, the lower the generalization error. The margin planes contain the examples of the classes that are nearest to each other. In the case of linearly not separable classes (when the classes overlap, see Figure 3.5), a soft-margin SVC may be employed. For examples on the correct side of the hyperplane, the loss is zero. For wrongly classified examples, the loss is proportional to the distance from the hyperplane. A global budget for loss is defined as a constraint and the hyperplane found subject to the constraint (Friedman, Hastie, and Tibshirani 2001).

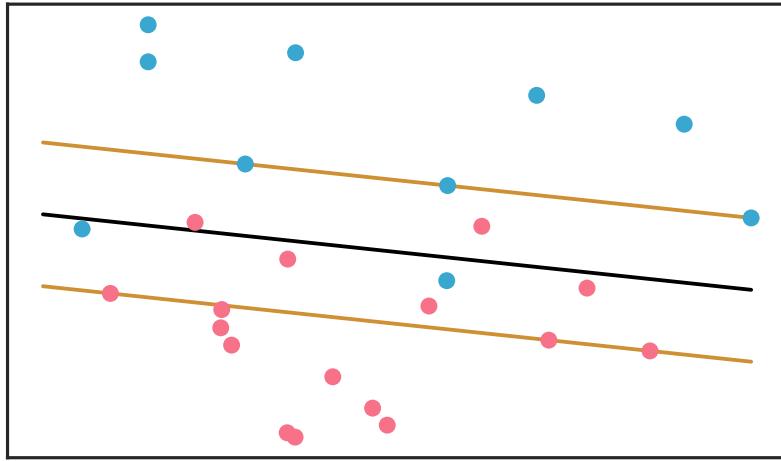


Figure 3.5.: Schematic display of an SVM hyperplane (in black), separating two overlapping classes. The margins are shown around the hyperplane, with support vectors falling on the margins. Misclassifications (examples on the wrong side of the hyperplane) have a total budget for distance from the separating plane. The margins are determined by respecting the budget. Adapted from Friedman, Hastie, and Tibshirani (2001).

Support Vector Machines (SVM), (Boser, Guyon, and Vapnik 1992; Cortes and Vapnik 1995) are another approach to the overlapping class case. By mapping the original input space to a high- or infinite-dimensional feature space using nonlinear transformations, a linear hyperplane separating the classes can be found. Calculation of the SVM involves a dot product between the x_i (Friedman, Hastie, and Tibshirani 2001):

$$f(x) = \sum_{i=1}^N \alpha_i * y_i \langle h(x), h(x_i) \rangle + \beta_0 \quad (3.18)$$

where $h(x)$ are the transformations mapping from the input space to the high-dimensional space.

3. Experimental Setup and Methods

Since the transformations can be prohibitively expensive, the so-called kernel trick (Aizerman, Braverman, and Rozoner 1964) is used (Boser, Guyon, and Vapnik 1992). The trick lies in the fact that knowledge of the kernel function $K(x, x') = \langle h(x), h(x') \rangle$ suffices, without need to compute the dot products.

Several kernels are possible. In `scikit-learn`, linear, polynomial, radial basis function and sigmoid are available.

For regression, the Support Vector Regression Machine (SVR) was introduced by Drucker et al. (1997).

`scikit-learn` implementations were used for both classification and regression.

Hyperparameters

- c , $(0, \inf)$: Penalty on the margin size
- `kernel`: One of linear, poly, rbf or sigmoid
- `degree` $\{1, 2, \dots\}$: Degree of polynomial kernel
- `class_weight`: Automatic balancing or a dictionary with weights per class

3.5.4.6. Bayesian Ridge Regression

Bayesian Ridge Regression (BR) can be seen as a Bayesian approach to a linear model with l_2 regularization. The `sklearn.linear_model.BayesianRidge` algorithm was used, which is implemented as described in Tipping (2001). The linear model is:

$$y_i = f(x_n, w) + \epsilon_i \quad (3.19)$$

where ϵ_i are iid error terms $\sim \mathcal{N}(0, \sigma^2)$.

The probabilistic model is then:

$$p(y|x) = \mathcal{N}(y|f(x, w), \alpha) \quad (3.20)$$

In order to regularize the model in a ridge manner, a prior probability on the weights in $f(x, w)$ is defined as:

$$p(w|\lambda) = \prod_{i=0}^p \mathcal{N}(w_i|0, \lambda_i^{-1}) \quad (3.21)$$

The algorithm was trained with default settings.

4. Results and Discussion

Below, the results from data preprocessing, model evaluation/-selection and predictions are shown.

4.1. Preprocessing With Package kdd98

The self-written package `kdd98` ensures consistent data provisioning. It handles downloading and pre-processing of the data set for both the learning and validation set. All preprocessing steps are trained on the learning data set. The individual trained transformations are persisted on disk. After training, the transformations can be applied on the validation data. This process is transparent to the user. It is enough to instantiate the data provider for either learning or validation data set and request the data. Examples for usage can be found in the Jupyter notebooks.

The data sets can be obtained at intermediate steps from `kdd98.data_handler.KDD98DataProvider`:

- **raw**, as imported through `pandas.read_csv()`
- **preprocessed**, input errors removed, correct data types for all features, missing at random (MAR) imputations applied
- **numeric**, after feature engineering (encoded categories, date and zip code transformations)
- **imputed**, with missing values imputed
- **all-relevant**, filtered down to a set of relevant features

For some transformers, behavior can be controlled by specifying parameters (which has to be done in the code). The package's architecture makes it easy to implement additional transformation steps.

The source code, along with a short introduction, is available online¹.

4.2. Imputation

The evaluation of several imputation strategies led to a straightforward approach: Categorical features had a *missing* level added during encoding (Section 3.3.2). All other features were imputed by their median value to account for the skewed distributions. The notebook `4_Imputation.ipynb`² contains details on the other approaches studied.

In concordance with the cup documentation's requirements (Appendix B.1), constant and *sparse* features were dropped from the data set before imputation. The approach used in the R-package `caret` was implemented in a scikit-learn transformer for this purpose. Work from a blog post (Goddard 2017) was

¹<https://github.com/datarian/thesis-msc-statistics/tree/master/code/kdd98>

²https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/4_Imputation.ipynb

4. Results and Discussion

adapted. The beauty of the method used in caret is that it is data type agnostic. It works on the number of unique values per feature and the frequency ratio between the top and second values by count.

The features with most and least missing values are shown in Figure 4.1. It is not surprising to find the *MONTHS_TO_DONATION_** features among those with most missing because few examples respond to the promotions with a donation.

Among the incomplete features with least missing values, there are several of the US census features. The *RFA_** features give the status in reference to promotion *i*. All examples who did donate at some point before have an RFA status. Thus, the missingness indicates when new members were added because newly added members do not have an RFA status yet.

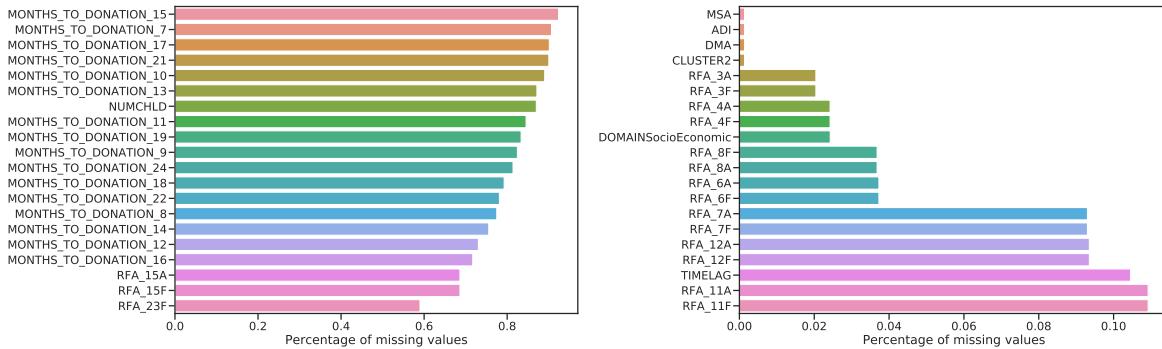


Figure 4.1.: Features with most (left) and fewest (right) missing values.

4.3. Feature Selection

After preprocessing and feature engineering, 653 features were present in the data. Using boruta, 58 features were identified as important, resulting in a 91 % reduction of the number of features. For details, refer to notebook *5_Feature_Extraction.ipynb*³.

Three groups of features were selected. By crosschecking with the dataset dictionary (Appendix B.2), their meaning could be identified:

Features from the **giving history** broadly correspond to those used in classical RFM models mentioned in literature (Section 1). It is reassuring to find them among the all-relevant features:

- Donation amount for promotion 14
- Summary features: All-time donation amount, all-time number of donations, smallest, average and largest donation, donation amount of most recent donation
- 24 Features on frequency and amount of donations as per the date of past donations
- Time since first donation, Time since largest donation, time since last donation
- Number of donations in response to card promotions
- Number of months between first and second donation
- An indicator for *star* donors

³https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/5_Feature_Extraction.ipynb

4. Results and Discussion

The **promotion history** features can be interpreted as a measure of the importance of the examples to the organization. Those who receive many promotions are deemed valuable:

- Number of promotions received
- Number promotions in last 12 months before the current promotion
- Number of card promotions received
- Number of card promotions in last 12 months before the current promotion

Features from the **US census** data are concerned with the social status and wealth of the neighborhood of donors and by intuition make sense to be deemed relevant.

- Median and average home value
- Percentage of home values above some threshold (5 features)
- Percentage of renters paying more than 500 \$
- DMA (designated market area, a geographical grouping)
- Median / average family / house income
- Per capita income
- Percentage of households with interest, rental or dividend income
- Percentage of adults with a Bachelor's degree
- Percentage of people born in state of residence

4.4. Model Evaluation and Selection

Model evaluation and -selection may be studied in detail in notebook *6_Model_Evaluation_Selection.ipynb*⁴. As will be explained below, all classifiers performed rather weak and were highly influenced by the imbalance in the data. The best results were achieved when using SMOTE resampling. Random over-/undersampling and specifying class weights had an inferior effect on model performance.

Among the classifiers evaluated, GLMnet showed the best performance and was thus chosen.

For the regression models, RF outperformed the other models, although the differences were less pronounced compared to the classifier results.

4.4.1. Classifiers

During grid search, models were trained individually for best F1 and recall. The models trained for high recall had only slightly worse precision than those trained for high F1, but at the same time better recall scores (see Jupyter notebook *6_Model_Evaluation_Selection.ipynb*). Therefore, the recall-trained models were considered for selection of the classifier.

Evaluation was based on the recall scores, confusion matrices, receiver operating characteristic (ROC) indicating model performance through an area under the curve score (ROC-AUC) and precision-recall (PR) curves.

⁴https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/6_Model_Evaluation_Selection.ipynb

4. Results and Discussion

If only the recall score were considered, the decision would be obvious, as shown in Figure 4.2. SVM has ~74 % recall, with the next-best scores at 54 % and 53 %. However, it is important to also consider the false positives as those cost money and decrease net profit.

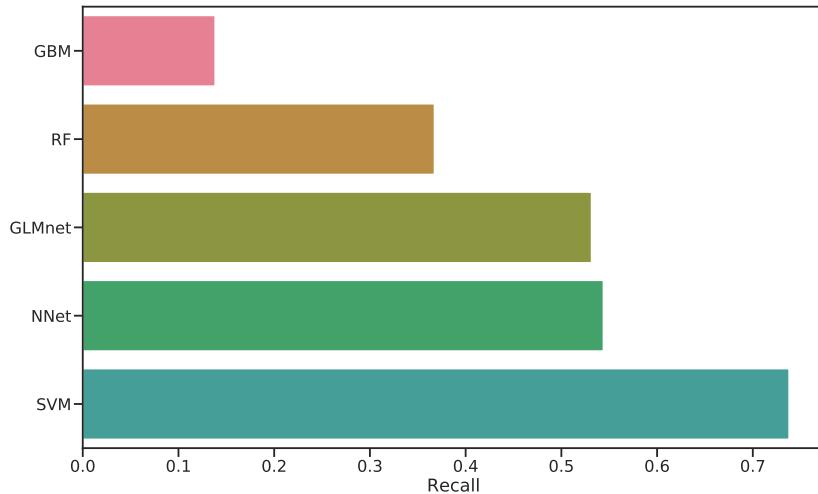


Figure 4.2.: Comparison of recall scores for all classifiers evaluated.

The ROC-AUC curve (Figure 4.3, left panel) was constructed by evaluating the false positive rate (FPR) against the true positive rate (TPR) at various thresholds for the predicted class probabilities of examples in the training data. The closer the curve is to the top-left corner, the better a model performs (large TPR and at the same time a low FPR for a wide range of thresholds). All classifiers seemingly performed rather weak. In the case of imbalanced data, the majority class dominates this metric. The false positive rate is $FPR = \frac{FP}{FP+TN}$. This means that as the false positives (FP) decrease due to an increasing threshold, FPR does not change a lot.

The PR curve with precision $P = \frac{TP}{TP+FP}$ plotted against recall $R = \frac{TP}{TP+FN}$ at different threshold values is sensitive to false positives and, since TN is not involved, better suited for the imbalanced data at hand. To construct the curve, recall is plotted against precision for various threshold values of the predicted class probabilities. For good models, the curve is close to the top-right corner. The right panel of Figure 4.3 shows the models in direct comparison. All of them suffer from low precision except for the highest threshold values. Again, this is caused by the high imbalance in the data.

More insight can be gained from the confusion matrices, shown in Figure 4.4. The best classifier should have a high recall, predicting many donors correctly, and at the same time a low False Positive Rate (FPR).

From the confusion matrices, it becomes obvious that for SVM, the trade off for high recall is also a high FPR. The false positives can be directly translated to cost: the 12'985 false positives in this case would amount to 8'830 \$ at a unit cost of 0.68 \$, while the 715 true positives generate an expected profit of only 8'808 \$ (with a mean net profit of 12.32 \$).

Evidently, GLMnet and NNet have a relatively good balance of recall and FPR. GBM performs well for FPR, which means less money lost due to unit costs, but has a very low recall.

GLMnet and NNet were also combined into a voting classifier. This classifier creates an ensemble that predicts through a majority vote, therefore compensating for the individual classifier's weaknesses.

4. Results and Discussion

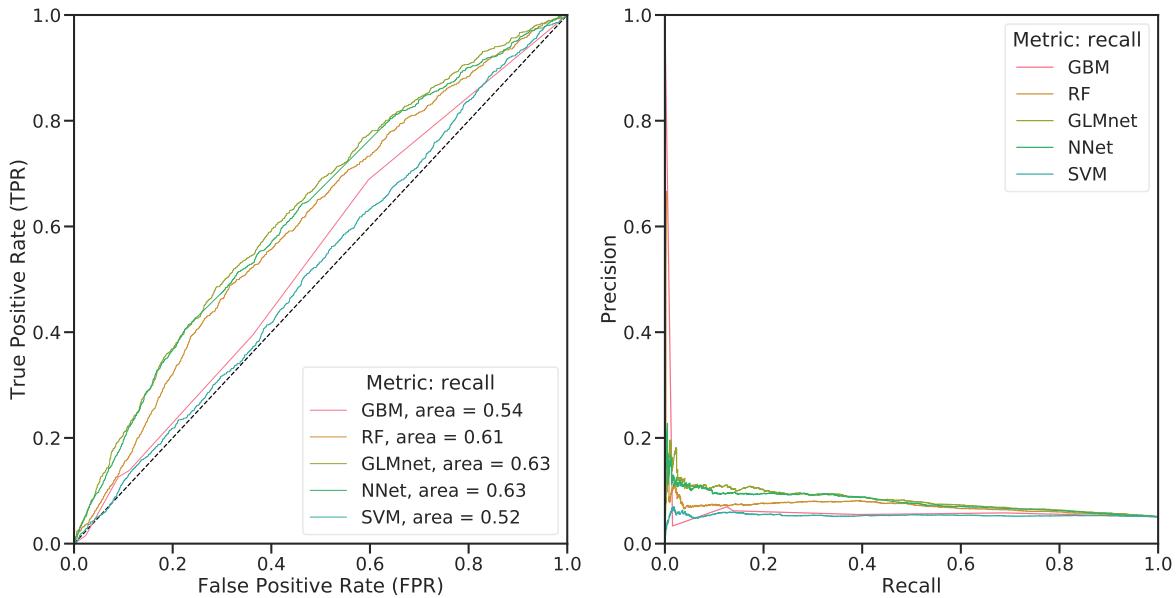


Figure 4.3.: Comparison of ROC-AUC for the evaluated classifiers (left) and PR curves (right) for the classifiers.

It exhibits a slightly lower recall for a slight decrease in FPR. Since *recall* was seen as being more important, it was not investigated further.

The non-null coefficients of the GLMnet model are shown in 4.5. 22 coefficients are non-null. The absolute values of the coefficients indicate importance of the respective feature. The first five features are from the promotion- and giving history and are concerned with the amount and frequency of donations.

Although RF performed poorly, it can be used to have a second source of information on the important features. The measure implemented in scikit-learn is the *Gini importance* as described in Breiman et al. (1984). It represents the total decrease in impurity (see Section 3.5.4.1) due to nodes split on feature f , averaged over all trees in the forest. The most important features again are all from the giving history, followed by the promotion history. US census features are not important (see Figure 4.6).

4. Results and Discussion

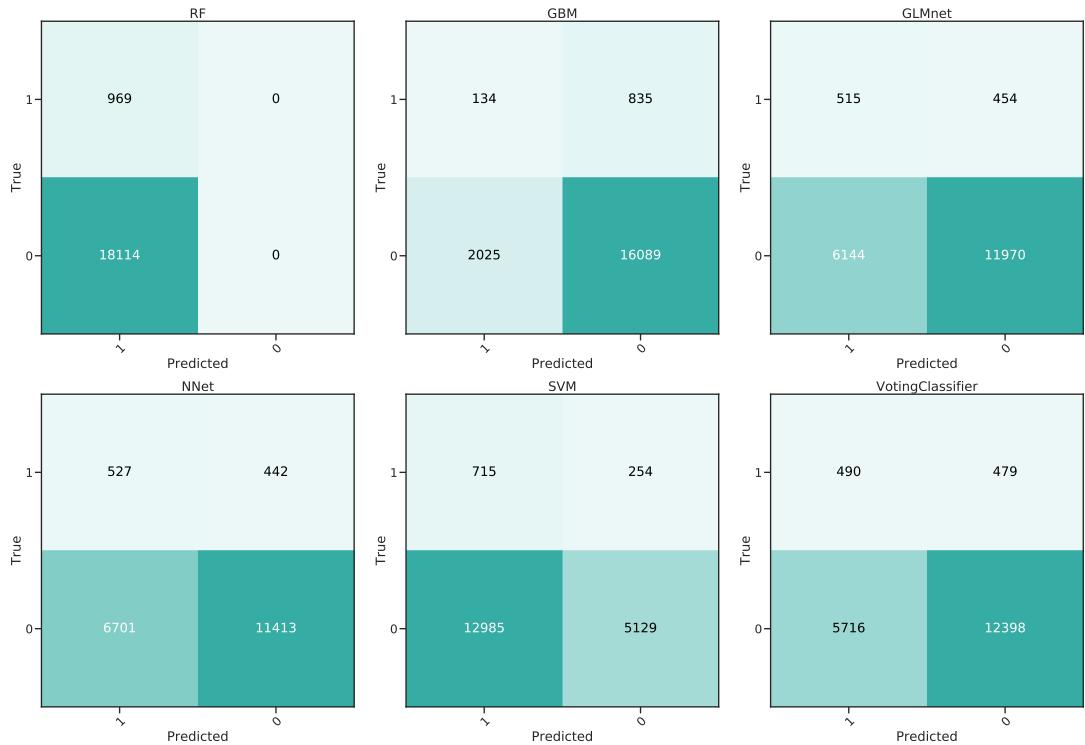


Figure 4.4.: Confusion matrices for the 6 classifiers studied.

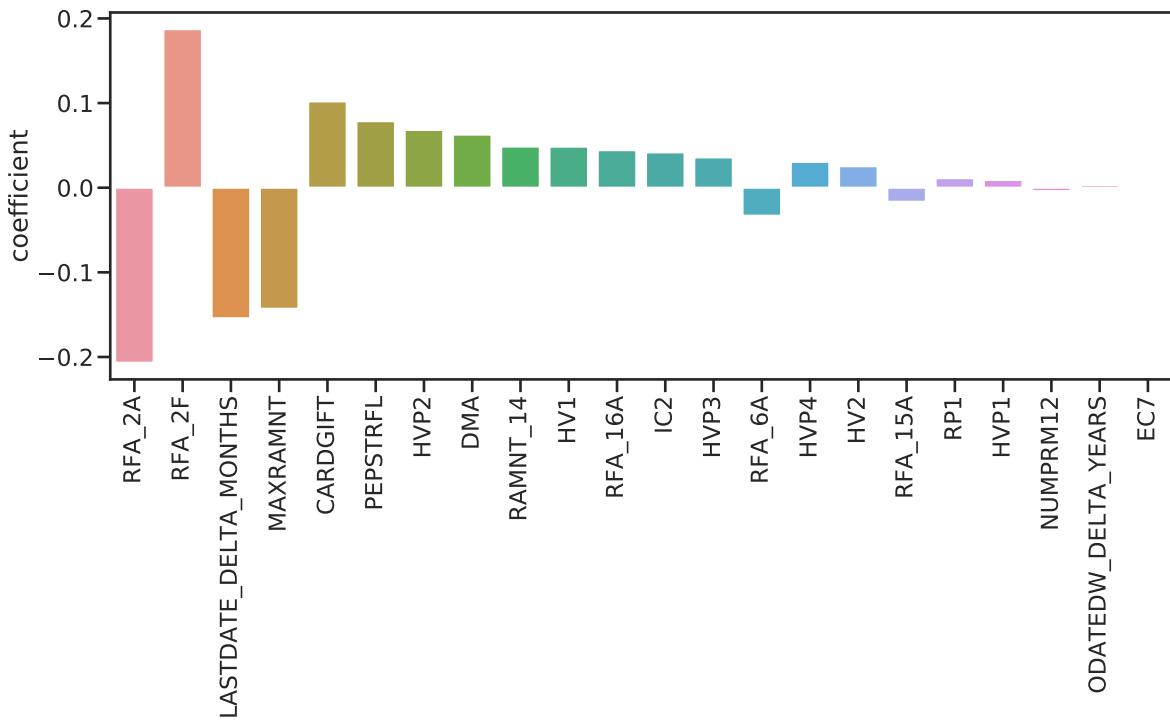


Figure 4.5.: Coefficient values for GLMnet.

4. Results and Discussion

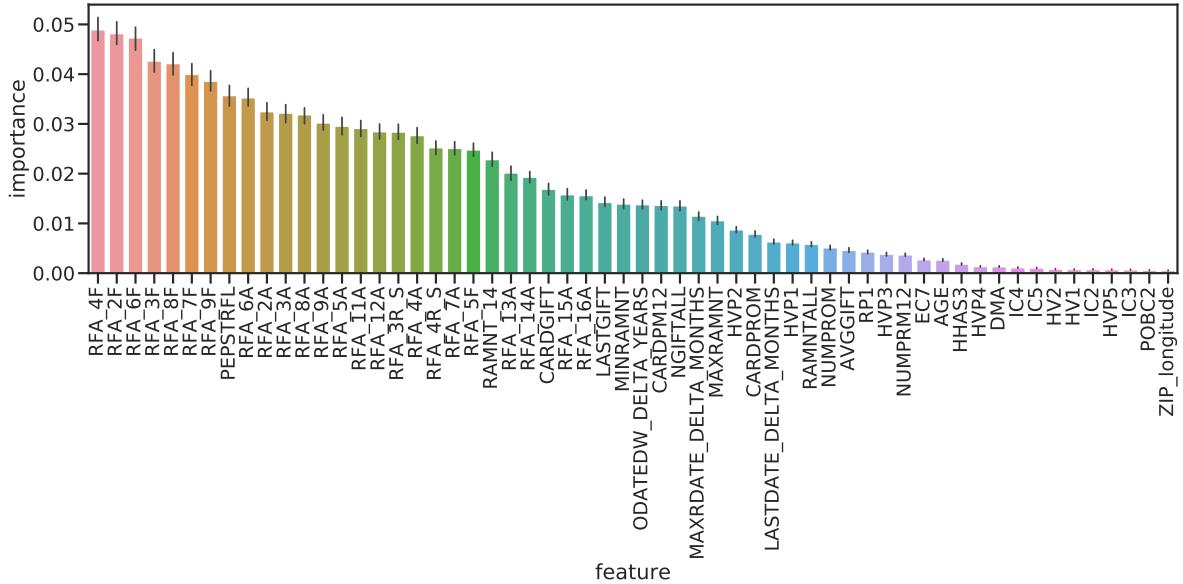


Figure 4.6.: Feature importances determined with the RF classifier. Impurity is measured by Gini importance / mean decrease impurity. Error bars give bootstrap error on 50 repetitions.

4.4.2. Regressors

As explained in Section 3.4, the regressors were learned on a subset of the training data comprised of all donors: $\{(x_i, y_i) | y_{b,i} = 1\}$. Before learning, $TARGET_D$ was Box-Cox transformed for normalization with $\lambda = 0.0239$. The goal of the transformation was to improve regression models' performance. The transformed data somewhat resembles a normal distribution, although there are several modes to be made out (see Figure 4.7).

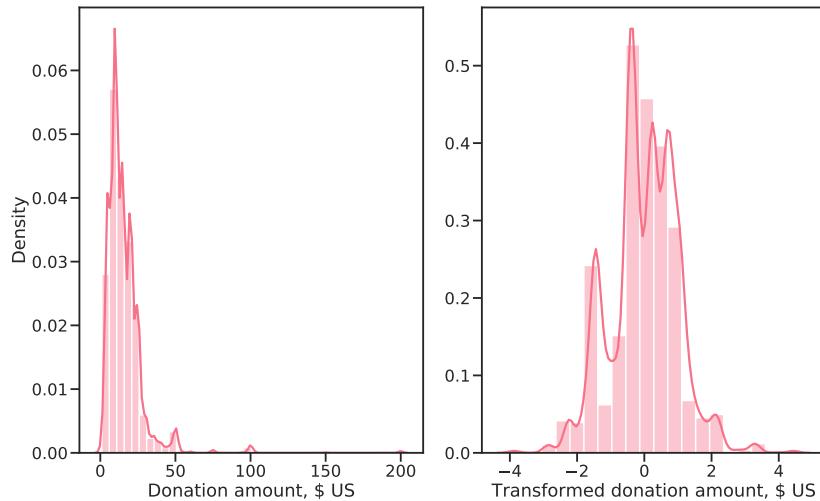


Figure 4.7.: Target before transformation (left) and after a Box-Cox transformation (right).

The resulting distribution of predicted donation amounts on the training data is shown in Figure 4.8. Except for SVR, the models produce very similar results. Again, the multi-modal distribution is found

4. Results and Discussion

(refer to Figure 4.7). RF and SVR are relatively symmetric, while BR and ElasticNet produce right-skewed distributions that predict very large donation amounts for some examples.

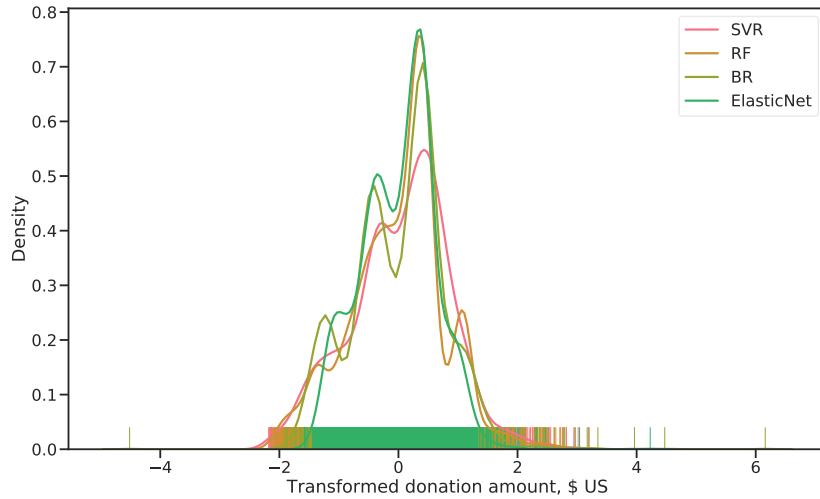


Figure 4.8.: Distribution of (Box-Cox transformed) donation amounts for the four regressors evaluated.

The regressor for final prediction was selected by R^2 score on a test set (20% of the learning data was used). RF was the best performing model with $R^2 = 0.72$ (see Figure 4.9).

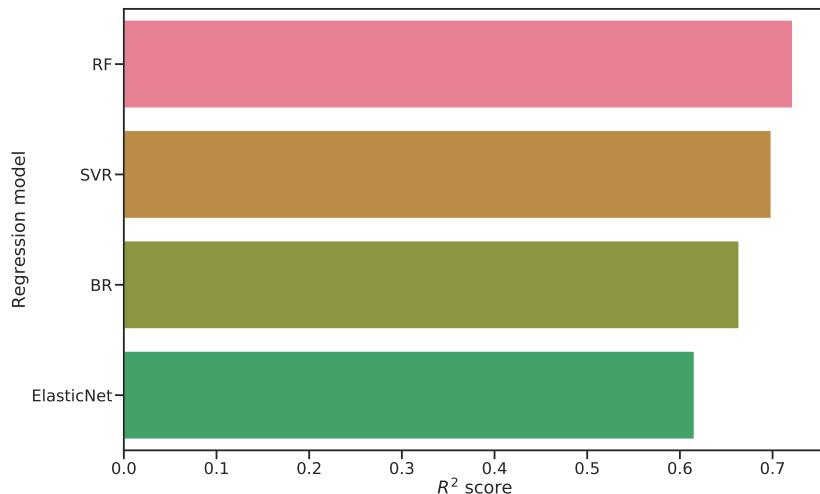


Figure 4.9.: Evaluation metric R^2 for all regression models evaluated. The domain for R^2 is $(-\infty, 1]$.

Again, RF enables to interpret the importance of features. The results (Figure 4.10) show that mainly the amount of the last donation, and to a lesser amount the average donation amount and the amounts of the last three donations were important for predicting donation amounts. This would indicate that donors tend to always donate the same sums.

4. Results and Discussion

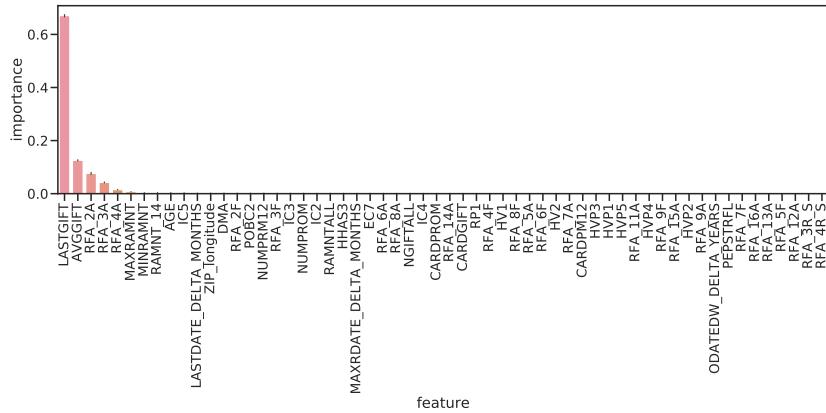


Figure 4.10.: Feature importances for the RF regressor.

4.5. Prediction

The intermediate steps for arriving at the final prediction will be examined in this section. For details on the steps, see `7_Predictions.ipynb`⁵. Up until the final prediction, the results reported are on the learning data set. The test data was only used for the final prediction for comparison with the results of the participants of the cup.

4.5.1. Prediction of Donation Probability

Using the GLMnet regressor on the learning data set, \hat{y}_b is distributed close to normal with a slight right skew (see Figure 4.11). 33'559 examples are selected, which amounts to 35.2 % of the training data set.

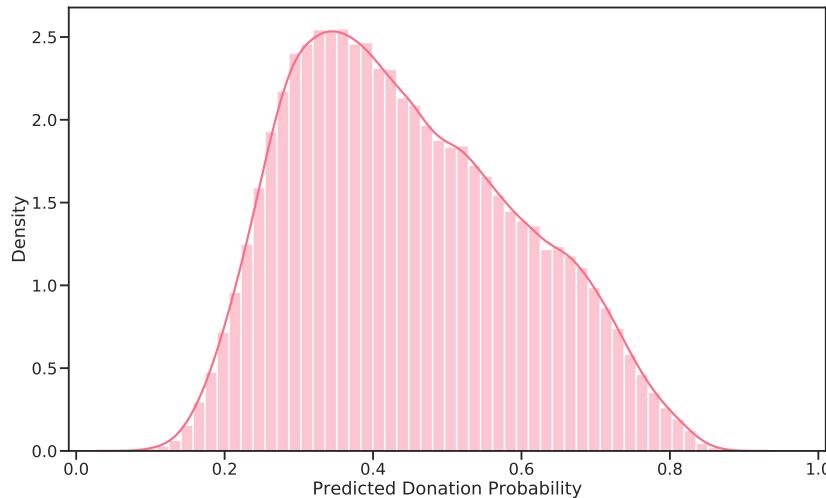


Figure 4.11.: Distribution of predicted donation probabilities \hat{y}_b on the learning data set.

⁵https://github.com/datarian/thesis-msc-statistics/tree/master/code/notebooks/7_Predictions.ipynb

4. Results and Discussion

4.5.2. Conditional Prediction of the Donation Amount

Compared to the true distribution of y_d (see Figure 4.7), a similar distribution for \hat{y}_d results from predictions with the RF regressor on the complete learning data, shown in Figure 4.12. The distribution is again multi-modal, with approximately the same median value. The predicted donation amounts are however strictly positive with a minimum of 3.2 \$. The reason is that the predictions are biased due to the non-random sample of donors used to learn the regressor. Unfortunately, the high-dollar donations are missing. The highest prediction is 82.1 \$, the median is 30.7 \$, which is almost double the median of the complete learning data set.

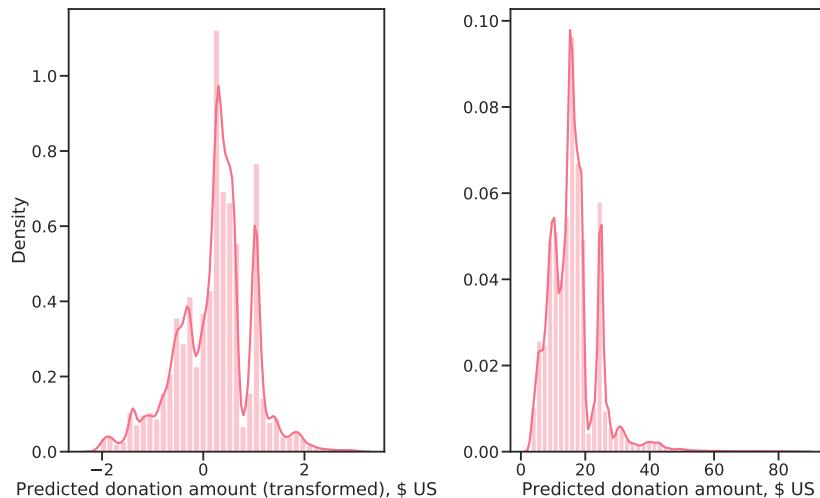


Figure 4.12.: Conditionally predicted donation amounts, Box-Cox transformed (left) and on the original scale (right).

4.5.3. Profit Optimization

The correction factor α , used to account for bias introduced by learning the regression models on a non-random sample, was found as described in section 3.4. First, estimated net profit $\hat{\Pi}$ was calculated using equation (3.4) for a grid of α values in $[0, 1]$. Then, a cubic spline was fitted to the data and α optimized subsequently.

As can be observed in Figure 4.13, the region of α for high profit is narrow. This is not surprising given the distribution of \hat{y}_d (Figure 4.12), which is narrowly concentrated around ~15 \$. Furthermore, the curve is constant over much of the domain, meaning that all examples were selected from $\alpha \approx 0.15$. The cubic spline fits the data very well and finding the maximum at $\alpha^* = 0.02$. For reference, a polynomial of degree 12 is also shown in 4.13, highlighting the difference in the fit for the two approaches.

4. Results and Discussion

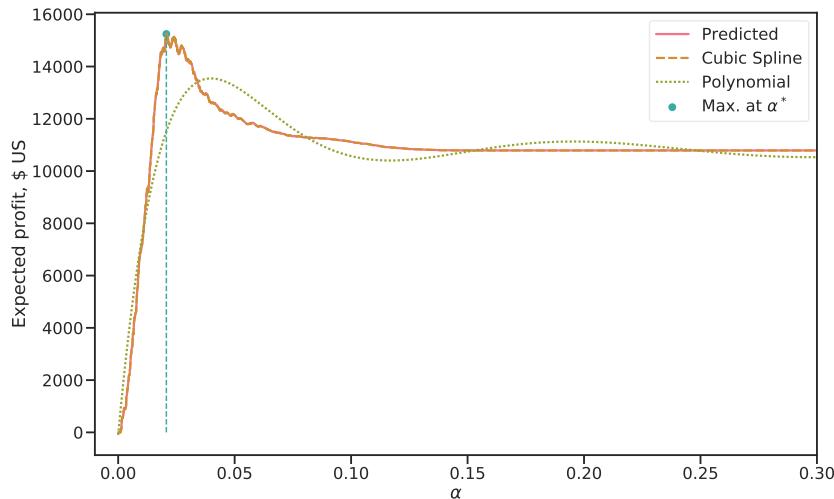


Figure 4.13.: Expected profit for a range of α values in $[0, 1]$ with overlayed cubic spline and polynomial function of order 12.

4.5.4. Final Prediction

For the final prediction, an estimator combining the two-step prediction process was implemented in package `kdd98`. The estimator `Kdd98ProfitEstimator` was initialized with the `GLMnet` classifier and `RF` regressor determined as the best during model evaluation and selection.

Then, the estimator was fitted on the complete learning data set, thereby fitting both the classifier and regressor and optimizing α^* , which was computed as $\alpha^* = 0.023$.

The fitted estimator could then be used for prediction on the test data set. The results are shown in Table 4.1 together with the winners of the cup. The model learned in this thesis selects less donors compared to the top-ranked participants and has a higher mean donation amount. Nevertheless, net profit is lower, resulting in the 4th rank.

The winner, Urban Science Applications with their proprietary software Gainsmarts, chose a two-step approach as well. In the first step, a logistic regression was used. The second step consisted of linear regression. Their software automated feature-engineering (by trying several different transformations for each feature) and feature selection through an expert system⁶. They selected a distinct subset of features for each step.

No information is available on SAS's approach. Decisionhouse's Quadstone on 3rd rank also invested heavily in feature engineering. The proprietary software was specially designed for customer behavior modeling. They used decision trees for conditioned profit estimation using 6 features and an additive scorecard model for selecting examples based on 10 features in combination. For the estimation of donation amount, they also found the amount of the last donation and average donation amounts as important.

Surely, if domain knowledge had influenced feature engineering and feature selection for the model developed here, performance would have been better. Given the radically data-driven approach, the result can be considered good.

⁶see <https://www.kdnuggets.com/meetings-past/kdd98/gain-kddcup98-release.html>

4. Results and Discussion

Table 4.1.: Prediction results for the test data set (in color) and the results of the cup-winners. N^* denotes number of examples selected. The theoretical maximum was calculated with: $\$ (\sum_{i=1}^n \mathbb{1}_{\{\text{TARGET}_{D,i} > 0.0\}} * (\text{TARGET}_{D,i} - u))$ with u the unit cost of 0.68 \\$ per mailing).

	Amount, \\$						
	N*	Min	Mean	Std	Max	Net Profit	Percent of Maximum
GainSmarts	56330	-0.68	0.26	5.57	499.32	14712	20.22
SAS	55838	-0.68	0.26	5.64	499.32	14662	20.15
Quadstone	57836	-0.68	0.24	5.66	499.32	13954	19.17
Own	40984	-0.68	0.34	6.28	499.32	13877	19.07
CARRL	55650	-0.68	0.25	5.61	499.32	13825	19.00

However, these predictions were made using the true TARGET_D . When treating the data as unseen, predicted net profit was unrealistically high, at 861'562 \\$, which translates to an error of 6'208 %. In this case, net profit was calculated by substituting $(y_d - u)$ with $(\hat{y}_d - u)$ in Equation (3.4). However bad this result is, would the model developed be used to select examples for a promotion, it would still lead to relatively good net profit.

5. Conclusions

- The effort put into preprocessing and development of the package `kdd98` laid a basis for future work on the data set. The architecture of the package makes modifications to existing transformers or implementation of additional transformations time-efficient. Care was taken to document the code so that future users can quickly add their own modifications.
- Regarding model learning, the current approach using a two-step prediction is promising in principle. Shortcomings in terms of classifier and regressor performance are suspected to mainly result from the remaining high dimensionality, even after reducing the number features by 90%. A logical next step would be to reiterate over feature engineering, imputation, and feature selection:
 - In feature engineering, the promotion- and giving history features should be carefully reexamined. Given the selection of various *RFA_* features for promotions in the past through `boruta`, these could be distilled into single *R*, *F*, *A* summaries.
 - Imputation with a CART method like in R’s `caret` package is another imputation strategy to consider. The simple strategy implemented in this thesis might have led to bias during model learning.
 - Feature selection using `boruta` could be combined with a manual step afterwards, informed by domain knowledge.
- The 4th rank achieved proves that selection of examples works in principle. The high error on predicted net profit however suggests a thorough reconsideration of the prediction method is necessary. Again, the implementation in `kdd98` can serve as a basis, providing the necessary “plumbing” for efficient examination and future work.
- The result in comparison with the performance of specialized software coupled with expertise in the field employed by the cup winners emphasizes the power of domain knowledge. The progress made in machine learning over the past 20 years since the cup does not automatically lead to superior results when strictly taking only the information contained in the data itself.

References

- Aizerman, Mark A, Emmanuel M Braverman, and LI Rozoner. 1964. “Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning.” *Automation and Remote Control* 25: 821–37.
- Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, Rob Hyndman, and Ruben Arslan. 2018. *Rmarkdown: Dynamic Documents for R*. Chapman; Hall/CRC.
- Bellman, Richard E. 1966. “Dynamic Programming.” *Science* 153 (3731): 34–37.
- Bergstra, James, and Yoshua Bengio. 2012. “Random Search for Hyper-Parameter Optimization.” *Journal of Machine Learning Research* 13 (Feb): 281–305.
- Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik. 1992. “A Training Algorithm for Optimal Margin Classifiers.” In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–52. COLT ’92. New York, NY, USA: ACM.
- Breiman, Leo. 1996. “Bagging Predictors.” *Machine Learning* 24 (2): 123–40.
- . 2001. “Random Forests.” *Machine Learning* 45 (January): 5–32.
- Breiman, Leo, JH Friedman, R Olshen, and CJ Stone. 1984. “Classification and Regression Trees.”
- Chen, Tianqi, and Carlos Guestrin. 2016. “Xgboost: A Scalable Tree Boosting System.” In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–94. ACM.
- Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20 (3): 273–97.
- Drucker, Harris, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. “Support Vector Regression Machines.” In *Advances in Neural Information Processing Systems*, 155–61.
- Fader, Peter S, Bruce GS Hardie, and Ka Lok Lee. 2005. “RFM and Clv: Using Iso-Value Curves for Customer Base Analysis.” *Journal of Marketing Research* 42 (4): 415–30.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. 10. Springer series in statistics New York, NY, USA.
- Goddard, Philip. 2017. “Machine Learning Pipelines with Scikit-learn.”
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

5. Conclusions

- Hahnloser, Richard HR, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. 2000. “Digital Selection and Analogue Amplification Coexist in a Cortex-Inspired Silicon Circuit.” *Nature* 405 (6789): 947.
- Hastie, Trevor, and Junyang Qian. 2014. “Glmnet Vignette.” *Retrieve from Http://Www. Web. Stanford. Edu/~ Hastie/Papers/Glmnet_Vignette.pdf. Accessed 30.05.2019* 20: 2016.
- Heckman, James J. 1976. “The Common Structure of Statistical Models of Truncation, Sample Selection and Limited Dependent Variables and a Simple Estimator for Such Models.” In *Annals of Economic and Social Measurement, Volume 5, Number 4*, 475–92. NBER.
- Hughes, AM. 1996. “Boosting Response with Rfm.” *Marketing Tools* 5 (January): 4–7.
- Hughes, Gordon. 1968. “On the Mean Accuracy of Statistical Pattern Recognizers.” *IEEE Transactions on Information Theory* 14 (1): 55–63.
- Hunter, JD. 2007. “Matplotlib: A 2D Graphics Environment.” *Computing in Science Engineering* 9 (3): 90–95.
- Kearns, Michael. 1988. “Thoughts on Hypothesis Boosting.” *Unpublished Manuscript* 45: 105.
- Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. “Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows.” Edited by F Loizides and B Schmidt. IOS Press.
- Kohavi, Ron, and others. 1995. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.” In *Ijcai*, 14:1137–45. 2. Montreal, Canada.
- Kohavi, Ron, and Rajesh Parekh. 2004. “Visualizing Rfm Segmentation.” In *Proceedings of the 2004 Siam International Conference on Data Mining*, 391–99. SIAM.
- Kursa, Miron B, and Witold Rudnicki. 2011. “The All Relevant Feature Selection Using Random Forest.” *arXiv Preprint arXiv:1106.5112*, June.
- Kursa, Miron B, Witold R Rudnicki, and others. 2010. “Feature Selection with the Boruta Package.” *Journal of Statistical Software* 36 (11): 1–13.
- Lemaître, Guillaume, Fernando Nogueira, and Christos K Aridas. 2017. “Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.” *Journal of Machine Learning Research* 18 (17): 1–5.
- McCarty, John A, and Manoj Hastak. 2007. “Segmentation Approaches in Data-Mining: A Comparison of Rfm, Chaid, and Logistic Regression.” *Journal of Business Research* 60 (6): 656–62.
- McKinney, Wes. 2010. “Data Structures for Statistical Computing in Python.” In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 51–56.
- . 2011. “Pandas: A Foundational Python Library for Data Analysis and Statistics.” *Python for High Performance and Scientific Computing* 14.
- Oliphant, Travis E. 2006. *A Guide to Numpy*. Vol. 1. Trelgol Publishing USA.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30.
- Rubin, Donald B. 1976. “Inference and Missing Data.” *Biometrika* 63 (3): 581–92.

5. Conclusions

- Stubseid, Saavi, and Ognjen Arandjelovic. 2018. “Machine Learning Based Prediction of Consumer Purchasing Decisions: The Evidence and Its Significance.” In *AAAI Conference on Artificial Intelligence*.
- Tipping, Michael E. 2001. “Sparse Bayesian Learning and the Relevance Vector Machine.” *Journal of Machine Learning Research* 1 (Jun): 211–44.
- Troyanskaya, Olga, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. 2001. “Missing Value Estimation Methods for Dna Microarrays.” *Bioinformatics* 17 (6): 520–25.
- van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. “Mice: Multivariate Imputation by Chained Equations in R.” *Journal of Statistical Software, Articles* 45 (3): 1–67.
- Waskom, Michael, O Botvinnik, P Hobson, J Warmenhoven, JB Cole, Y Halchenko, J Vanderplas, et al. 2014. “Seaborn: Statistical Data Visualization.” *Seaborn: Statistical Data Visualization Seaborn 0 5*.
- Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Chapman and Hall/CRC.
- . 2016. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC.
- Yeh, I-Cheng, King-Jang Yang, and Tao-Ming” Ting. 2009. “Knowledge Discovery on Rfm Model Using Bernoulli Sequence.” *Expert Systems with Applications* 36 (3, Part 2): 5866–71.
- Zou, Hui, and Trevor Hastie. 2005. “Regularization and Variable Selection via the Elastic Net.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2): 301–20.

A. Software

A.1. Python Environment

Python was installed through the Anaconda distribution¹.

The analysis documented should be reproduced using the same environment, thereby ensuring that the packages produce the same results and expose the API's used in the code.

The environment can be created using the file `environment.yml` in the main repository directory at <https://github.com/datarian/thesis-msc-statistics/tree/master/code/>.

Once conda is installed the following two commands first set up the environment with all the analysis' required packages, then activates that environment. Only after running these commands should the package `kdd98` be installed and the notebooks evaluated.

```
> conda env create -f environment.yml  
> source activate msc-thesis-fh
```

A.2. Package `kdd98`

The source code is at: <https://github.com/datarian/thesis-msc-statistics/tree/master/code/kdd98>

A.2.1. Usage

A.2.1.1. Data Provisioning

```
from kdd98 import data_handler as dh

provider = dh.KDD98DataProvider(["cup98LRN.txt", "cup98VAL.txt"])

# Get data at several intermediate steps as necessary
provider.raw_data
provider.preprocessed_data
provider.numeric_data
provider.imputed_data
provider.all_relevant_data
```

By requesting `all_relevant_data`, all transformation steps are applied. Request data for `cup98LRN.txt` first to learn all transformers involved. Only then can the provider be instantiated with `cup98VAL.txt` and the exact same transformations applied to that data set.

¹Available from <https://www.anaconda.com>

A. Software

A.2.1.2. Predictions

```
from kdd98.prediction import Kdd98ProfitEstimator

# These are found beforehand and expected to implement scikit-learn's API
classifier = SomeClassifier() # predicts donation probability
regressor = SomeRegressor() # conditionally predicts donation amount

estimator = Kdd98ProfitEstimator(classifier, regressor)

estimator.fit(X_learn, y_learn)

# Returns a tuple (indicator, net_profit):
estimator.predict(X_test, y_test)
```

A.2.2. Installation

Clone the repository, then, from the base directory:

```
cd code

python setup.py install # for direct install

# for a conda package to be installed manually afterwards:
python setup.py bdist_conda
```

A.2.2.1. Project structure

After installing the package, several folders and files are created in the working directory when using it:

- data contains the data files and documentation of the original KDD Cup 1998
- data/data_frames contains pickled pandas df's with data at the corresponding transformation step
- models/internal contains persisted transformers for data set transformations.
- out.log reports transformation progress / potential problems

B. KDD Cup Documents

B.1. Cup Documentation

=====
EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL

INFORMATION LISTED BELOW IS AVAILABLE UNDER THE TERMS OF THE
CONFIDENTIALITY AGREEMENT

EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL
=====

```
+-----+  
|            DOCUMENTATION TO ACCOMPANY        |  
|            |  
|            KDD-CUP-98                            |  
|            |  
|            The Second International Knowledge Discovery and        |  
|            Data Mining Tools Competition        |  
|            |  
|            Held in Conjunction with KDD-98                        |  
|            |  
|            The Fourth International Conference on Knowledge        |  
|            Discovery and Data Mining                |  
|            [www.kdnuggets.com] or                    |  
|            [www-aig.jpl.nasa.gov/kdd98] or        |  
|            [www.aaai.org/Conferences/KDD/1998]        |  
|            |  
|            Sponsored by the                          |  
|            |  
|            American Association for Artificial Intelligence (AAAI)        |  
|            Epsilon Data Mining Laboratory            |  
|            Paralyzed Veterans of America (PVA)        |  
+-----+  
|            |  
|            Created: 7/20/98                        |  
|            Last update: 7/22/98                    |  
|            File name: cup98DOC.txt                |  
+-----+
```

Table of Contents:

- o IMPORTANT DATES (UPDATED)
- o GENERAL INSTRUCTIONS (for DOWNLOADS, RESULT RETURNS, etc.)
- o LISTING of the FILES (Contents of the README FILE)
- o PROJECT OVERVIEW: A FUND RAISING NET RETURN PREDICTION MODEL
- o EVALUATION RULES

B. KDD Cup Documents

- o DATA SOURCES and ORDER & TYPE OF THE VARIABLES IN THE DATA SETS
- o SUMMARY STATISTICS (MIN & MAX)
- o DATA (PRE)PROCESSING
- o KDD-CUP-98 PROGRAM COMMITTEE
- o TERMINOLOGY-GLOSSARY

```
+-----+  
| IMPORTANT DATES (UPDATED) |  
+-----+
```

- o Release of the datasets, related documentation and the KDD-CUP questionnaire

July 22, 1998

- o Return of the results and the KDD-CUP questionnaire

August 19, 1998

- o KDD-CUP Committee evaluation of the results

August 19-25

- o Individual performance evaluations send to the participants

August 26, 1998

- o Public announcement of the winners and awards presentation during KDD-98 in New York City

August 29, 1998

```
+-----+  
| GENERAL INSTRUCTIONS (for DOWNLOADS, RESULT RETURNS, etc.) |  
+-----+
```

1. FTP to 159.127.66.10. Login anonymous. Enter email ID as password.

3. The README file contains information about the files included in the FTP server. All data files are compressed. The files with .zip extension are compressed with the PKZIP compression utility and they are for participants with IBM PC compatible hardware. The PKUNZIP utility is needed to unzip these files. The files with .Z extension are UNIX COMPRESSED and they are for the participants with UNIX compatible hardware. YOU WILL EITHER NEED THE DATA FILES <cup98LRN.ZIP AND cup98VAL.ZIP> *OR* <cup98LRN.TXT.Z AND cup98VAL.TXT.Z>, BUT NOT BOTH. REMEMBER TO FTP THESE FILES IN BINARY MODE.

4. The data sets are in comma delimited format. The learning dataset <cup98LRN.txt> contains 95412 records and 481 fields. The first/header row of the data set contains the field names.

The validation dataset <cup98VAL.txt> contains 96367 records and 479 variables. The first/header row of the data set contains the field names.

B. KDD Cup Documents

THE RECORDS IN THE VALIDATION DATASET ARE IDENTICAL TO THE RECORDS IN THE LEARNING DATASET EXCEPT THAT THE VALUES FOR THE TARGET/DEPENDENT VARIABLES ARE MISSING (i.e., the fields TARGET_B and TARGET_D are not included in the validation data set.)

5. The data dictionary (for both the learning and the validation data set) is included in the file <cup98DIC.txt>. The fields in the data dictionary are ordered by the position of the fields in the learning data set. The dictionary for the validation data set is identical to the dictionary for the learning data set except the two target fields (target_B and target_D) are missing in the validation data set.

6. Blanks in the string (or character) variables/fields and periods in the numeric variables correspond to missing values.

7. Each record has a unique record identifier or index (field name: CONTROLN.) For each record, there are two target/dependent variables (field names: TARGET_B and TARGET_D). TARGET_B is a binary variable indicating whether or not the record responded to the promotion of interest ("97NK" mailing) while TARGET_D contains the donation amount (dollar) and is only observed for those that responded to the promotion.

8. THE DEADLINE HAS BEEN EXTENDED. You are required to return the questionnaire and the validation dataset of 96367 records by email to <iparsa@epsilon.com> by AUGUST 19, 1998.

Each record in the returned file should consist of the following two values:

- a. The unique record identifier or index (field name: CONTROLN)
- b. Predicted value of the donation (dollar) amount (for the target variable TARGET_D) for that record

You are also required to fill out the questionnaire (file name: <cup98QUE.txt>. The questionnaire is used to summarize in bullet points the data analytic techniques you've applied to the dataset.

9. Please send email to <iparsa@epsilon.com> when you download the files so we can keep you informed about anything necessary.

10. Under no circumstances should any participant contact Paralyzed Veterans of America (PVA) for any reason.

If you have any questions, please send email to <iparsa@epsilon.com.>

```
+-----+  
| FILES LISTING (README FILE) |  
+-----+
```

File Naming Conventions:

- o cup98 : KDD-CUP-98
- o QUE : QUEstionnaire
- o DOC : DOCUMENTation
- o DIC : DICTIONary

B. KDD Cup Documents

o LRN : LeaRNing data set
o VAL : VALIDation data set
o .txt : plain ascii text files
o .zip : PKZIP compressed files
o .txt.Z: UNIX COMPRESSED files

FILE NAME	DESCRIPTION

README	This list, listing the files in the FTP server and their contents.
cup98NDA.txt	The Non-Disclosure Agreement. MUST BE SIGNED BY ALL PARTICIPANTS AND MAILED BACK TO ISMAIL PARSA <iparsa@epsilon.com> BEFORE DOWNLOADING THE DATA SETS.
cup98DOC.txt	This file, an overview and pointer to more detailed information about the competition
cup98DIC.txt	Data dictionary to accompany the analysis data set.
cup98QUE.txt	KDD-CUP questionnaire. PARTICIPANTS ARE REQUIRED TO FILL-OUT THE QUESTIONNAIRE and turned in with the results.
cup98LRN.zip	PKZIP compressed raw LEARNING data set. Internal name: cup98LRN.txt File size: 36,468,735 bytes zipped. 117,167,952 bytes unzipped. Number of Records: 95412. Number of Fields: 481.
cup98VAL.zip	PKZIP compressed raw VALIDATION data set. Internal name: cup98VAL.txt File size: 36,763,018 bytes zipped. 117,943,347 bytes unzipped. Number of Records: 96367. Number of Fields: 479.
cup98LRN.txt.Z	UNIX COMPRESSED raw LEARNING data set. Internal name: cup98LRN.txt File size: 36,579,127 bytes compressed. 117,167,952 bytes uncompressed. Number of Records: 95412. Number of Fields: 481.
cup98VAL.txt.Z	UNIX COMPRESSED raw VALIDATION data set. Internal name: cup98VAL.txt File size: 36,903,761 bytes compressed. 117,943,347 bytes uncompressed. Number of Records: 96367. Number of Fields: 479.

+-----+
| PROJECT OVERVIEW: A Fund Raising Net Return Prediction Model |
+-----+

B. KDD Cup Documents

BACKGROUND AND OBJECTIVES

The data set for this year's Cup has been generously provided by the Paralyzed Veterans of America (PVA). PVA is a not-for-profit organization that provides programs and services for US veterans with spinal cord injuries or disease. With an in-house database of over 13 million donors, PVA is also one of the largest direct mail fund raisers in the country.

Participants in the '98 CUP will demonstrate the performance of their tool by analyzing the results of one of PVA's recent fund raising appeals. This mailing was sent to a total of 3.5 million PVA donors who were on the PVA database as of June 1997. Everyone included in this mailing had made at least one prior donation to PVA.

The mailing included a gift (or "premium") of personalized name & address labels plus an assortment of 10 note cards and envelopes. All of the donors who received this mailing were acquired by PVA through similar premium-oriented appeals such as this.

One group that is of particular interest to PVA is "Lapsed" donors. These are individuals who made their last donation to PVA 13 to 24 months ago. They represent an important group to PVA, since the longer someone goes without donating, the less likely they will be to give again. Therefore, recapture of these former donors is a critical aspect of PVA's fund raising efforts.

However, PVA has found that there is often an inverse correlation between likelihood to respond and the dollar amount of the gift, so a straight response model (a classification or discrimination task) will most likely net only very low dollar donors. High dollar donors will fall into the lower deciles, which would most likely be suppressed from future mailings. The lost revenue of these suppressed donors would then offset any gains due to the increased response rate of the low dollar donors.

Therefore, to improve the cost-effectiveness of future direct marketing efforts, PVA wishes to develop a model that will help them maximize the net revenue (a regression or estimation task) generated from future renewal mailings to Lapsed donors.

POPULATION

The population for this analysis will be Lapsed PVA donors who received the June '97 renewal mailing (appeal code "97NK"). Therefore, the analysis data set contains a subset of the total universe who received the mailing.

The analysis file includes all 191,779 Lapsed donors who received the mailing, with responders to the mailing marked with a flag in the TARGET_B field. The total dollar amount of each responder's gift is in the TARGET_D field.

The overall response rate for this direct mail promotion is 5.1%. The

B. KDD Cup Documents

distribution of the target fields in the learning and validation files is as follows:

Learning Data Set
Target Variable: Binary Indicator of Response to 97NK
Mailing

TARGET_B	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
0	90569	94.9	90569	94.9
1	4843	5.1	95412	100.0

Learning Data Set
Target Variable: Donation Amount (in \$) to 97NK Mailing

Variable	N	Mean	Minimum	Maximum
TARGET_D	95412	0.7930732	0	200.0000000

Validation Data Set
Target Variable: Binary Indicator of Response to 97NK
Mailing

TARGET_B	Frequency	Percent	Cumulative	Cumulative
			Frequency	Percent
0	91494	94.9	91494	94.9
1	4873	5.1	96367	100.0

Validation Data Set
Target Variable: Donation Amount (in \$) to 97NK Mailing

Variable	N	Mean	Minimum	Maximum
TARGET_D	96367	0.7895819	0	500.0000000

The average donation amount (in \$) among the responders is:

Learning Data Set
Target Variable: Donation Amount (in \$) to 97NK
Mailing

N	Mean	Minimum	Maximum
4843	15.6243444	1.0000000	200.0000000

Validation Data Set
Target Variable: Donation Amount (in \$) to 97NK

B. KDD Cup Documents

Mailing

N	Mean	Minimum	Maximum
4873	15.6145372	0.3200000	500.0000000

COST MATRIX

The package cost (including the mail cost) is \$0.68 per piece mailed.

ANALYSIS TIME FRAME AND REFERENCE DATE

The 97NK mailing was sent out on June 1997. All information included in the file (excluding the giving history date fields) is reflective of behavior prior to 6/97. This date may be used as the reference date in generating the "number of months since" or "time since" or "elapsed time" variables. The participants could also find the reference date information in the filed ADATE_2. This field contains the dates the 97NK promotion was mailed.

+-----+
| EVALUATION RULES |
+-----+

Once again, the objective of the analysis will be to maximize the net revenue generated from this mailing - a censored regression or estimation problem. The response variable is, thus, continuous (for the lack of a better common term.) Although we are releasing both the binary and the continuous versions of the target variable (TARGET_B and TARGET_D respectively), the program committee will use the predicted value of the donation (dollar) amount (for the target variable TARGET_D) in evaluating the results. So, returning the predicted value of the binary target variable TARGET_B and its associated probability/strength will not be sufficient.

The typical outcome of predictive modeling in database marketing is an estimate of the expected response/return per customer in the database. A marketer will mail to a customer so long as the expected return from an order exceeds the cost invested in generating the order, i.e., the cost of promotion. For our purpose, the package cost (including the mail cost) is \$0.68 per piece mailed.

KDD-CUP committee will evaluate the results based solely on the net revenue generated on the hold-out or validation sample.

The measure we will use is:

Sum (the actual donation amount - \$0.68) over all records for which the expected revenue (or predicted value of the donation) is over \$0.68.

B. KDD Cup Documents

This is a direct measure of profit. The winner will be the participant with the highest actual sum. The results will be rounded to the nearest 10 dollars.

```
+-----+  
| DATA SOURCES and ORDER & TYPE OF THE VARIABLES IN THE DATA SETS |  
+-----+
```

The dataset includes:

- o 24 months of detailed PVA promotion and giving history (covering the period 12 to 36 months prior to the "97NK" mailing)
- o A summary of the promotions sent to the donors over the most recent 12 months prior to the "97NK" mailing (by definition, none of these donors responded to any of these promotions)
- o Summary variables reflecting each donor's lifetime giving history (e.g., total # of donations prior to "97NK" mailing, total \$ amount of the donations, etc.)
- o Overlay demographics, including a mix of household and area level data
- o All other available data from the PVA database (e.g., date of first gift, state, origin source, etc.)

The fields are described in greater detail in the data dictionary file <filename: cup98DIC.txt>.

The name of the variables in the learning and validation data sets is included in each file as the top (header) record. For your information, they are listed below again (ordered by data set position) along with the field type information (Num: numeric, Char: string/character.)

Field Name	Type
ODATEDW	Num
OSOURCE	Char
TCODE	Num
STATE	Char
ZIP	Char
MAILCODE	Char
PVASTATE	Char
DOB	Num
NOEXCH	Char
RECINHSE	Char
RECP3	Char
RECPVG	Char
RECSWEEP	Char
MDMAUD	Char
DOMAIN	Char
CLUSTER	Char
AGE	Num
AGEFLAG	Char

B. KDD Cup Documents

HOMEOWNR	Char
CHILD03	Char
CHILD07	Char
CHILD12	Char
CHILD18	Char
NUMCHLD	Num
INCOME	Num
GENDER	Char
WEALTH1	Num
HIT	Num
MBCRAFT	Num
MBGARDEN	Num
MBBOOKS	Num
MBCOLECT	Num
MAGFAML	Num
MAGFEM	Num
MAGMALE	Num
PUBGARDN	Num
PUBCULIN	Num
PUBHLTH	Num
PUBDOITY	Num
PUBNEWFN	Num
PUBPHOTO	Num
PUBOPP	Num
DATASRCE	Char
MALEMILI	Num
MALEVET	Num
VIETVETS	Num
WWIIVETS	Num
LOCALGOV	Num
STATEGOV	Num
FEDGOV	Num
SOLP3	Char
SOLIH	Char
MAJOR	Char
WEALTH2	Num
GEOCODE	Char
COLLECT1	Char
VETERANS	Char
BIBLE	Char
CATLG	Char
HOME	Char
PETS	Char
CDPLAY	Char
STEREO	Char
PCOWNERS	Char
PHOTO	Char
CRAFTS	Char
FISHER	Char
GARDENIN	Char
BOATS	Char
WALKER	Char
KIDSTUFF	Char
CARDS	Char
PLATES	Char
LIFESRC	Char
PEPSTRFL	Char

B. KDD Cup Documents

POP901	Num
POP902	Num
POP903	Num
POP90C1	Num
POP90C2	Num
POP90C3	Num
POP90C4	Num
POP90C5	Num
ETH1	Num
ETH2	Num
ETH3	Num
ETH4	Num
ETH5	Num
ETH6	Num
ETH7	Num
ETH8	Num
ETH9	Num
ETH10	Num
ETH11	Num
ETH12	Num
ETH13	Num
ETH14	Num
ETH15	Num
ETH16	Num
AGE901	Num
AGE902	Num
AGE903	Num
AGE904	Num
AGE905	Num
AGE906	Num
AGE907	Num
CHIL1	Num
CHIL2	Num
CHIL3	Num
AGEC1	Num
AGEC2	Num
AGEC3	Num
AGEC4	Num
AGEC5	Num
AGEC6	Num
AGEC7	Num
CHILC1	Num
CHILC2	Num
CHILC3	Num
CHILC4	Num
CHILC5	Num
HHAGE1	Num
HHAGE2	Num
HHAGE3	Num
HHN1	Num
HHN2	Num
HHN3	Num
HHN4	Num
HHN5	Num
HHN6	Num
MARR1	Num
MARR2	Num

B. KDD Cup Documents

MARR3	Num
MARR4	Num
HHP1	Num
HHP2	Num
DW1	Num
DW2	Num
DW3	Num
DW4	Num
DW5	Num
DW6	Num
DW7	Num
DW8	Num
DW9	Num
HV1	Num
HV2	Num
HV3	Num
HV4	Num
HU1	Num
HU2	Num
HU3	Num
HU4	Num
HU5	Num
HHD1	Num
HHD2	Num
HHD3	Num
HHD4	Num
HHD5	Num
HHD6	Num
HHD7	Num
HHD8	Num
HHD9	Num
HHD10	Num
HHD11	Num
HHD12	Num
ETHC1	Num
ETHC2	Num
ETHC3	Num
ETHC4	Num
ETHC5	Num
ETHC6	Num
HVP1	Num
HVP2	Num
HVP3	Num
HVP4	Num
HVP5	Num
HVP6	Num
HUR1	Num
HUR2	Num
RHP1	Num
RHP2	Num
RHP3	Num
RHP4	Num
HUPA1	Num
HUPA2	Num
HUPA3	Num
HUPA4	Num
HUPA5	Num

B. KDD Cup Documents

HUPA6	Num
HUPA7	Num
RP1	Num
RP2	Num
RP3	Num
RP4	Num
MSA	Num
ADI	Num
DMA	Num
IC1	Num
IC2	Num
IC3	Num
IC4	Num
IC5	Num
IC6	Num
IC7	Num
IC8	Num
IC9	Num
IC10	Num
IC11	Num
IC12	Num
IC13	Num
IC14	Num
IC15	Num
IC16	Num
IC17	Num
IC18	Num
IC19	Num
IC20	Num
IC21	Num
IC22	Num
IC23	Num
HHAS1	Num
HHAS2	Num
HHAS3	Num
HHAS4	Num
MC1	Num
MC2	Num
MC3	Num
TPE1	Num
TPE2	Num
TPE3	Num
TPE4	Num
TPE5	Num
TPE6	Num
TPE7	Num
TPE8	Num
TPE9	Num
PEC1	Num
PEC2	Num
TPE10	Num
TPE11	Num
TPE12	Num
TPE13	Num
LFC1	Num
LFC2	Num
LFC3	Num

B. KDD Cup Documents

LFC4	Num
LFC5	Num
LFC6	Num
LFC7	Num
LFC8	Num
LFC9	Num
LFC10	Num
OCC1	Num
OCC2	Num
OCC3	Num
OCC4	Num
OCC5	Num
OCC6	Num
OCC7	Num
OCC8	Num
OCC9	Num
OCC10	Num
OCC11	Num
OCC12	Num
OCC13	Num
EIC1	Num
EIC2	Num
EIC3	Num
EIC4	Num
EIC5	Num
EIC6	Num
EIC7	Num
EIC8	Num
EIC9	Num
EIC10	Num
EIC11	Num
EIC12	Num
EIC13	Num
EIC14	Num
EIC15	Num
EIC16	Num
OEDC1	Num
OEDC2	Num
OEDC3	Num
OEDC4	Num
OEDC5	Num
OEDC6	Num
OEDC7	Num
EC1	Num
EC2	Num
EC3	Num
EC4	Num
EC5	Num
EC6	Num
EC7	Num
EC8	Num
SEC1	Num
SEC2	Num
SEC3	Num
SEC4	Num
SEC5	Num
AFC1	Num

B. KDD Cup Documents

AFC2	Num
AFC3	Num
AFC4	Num
AFC5	Num
AFC6	Num
VC1	Num
VC2	Num
VC3	Num
VC4	Num
ANC1	Num
ANC2	Num
ANC3	Num
ANC4	Num
ANC5	Num
ANC6	Num
ANC7	Num
ANC8	Num
ANC9	Num
ANC10	Num
ANC11	Num
ANC12	Num
ANC13	Num
ANC14	Num
ANC15	Num
POBC1	Num
POBC2	Num
LSC1	Num
LSC2	Num
LSC3	Num
LSC4	Num
VOC1	Num
VOC2	Num
VOC3	Num
HC1	Num
HC2	Num
HC3	Num
HC4	Num
HC5	Num
HC6	Num
HC7	Num
HC8	Num
HC9	Num
HC10	Num
HC11	Num
HC12	Num
HC13	Num
HC14	Num
HC15	Num
HC16	Num
HC17	Num
HC18	Num
HC19	Num
HC20	Num
HC21	Num
MHUC1	Num
MHUC2	Num
AC1	Num

B. KDD Cup Documents

AC2	Num
ADATE_2	Num
ADATE_3	Num
ADATE_4	Num
ADATE_5	Num
ADATE_6	Num
ADATE_7	Num
ADATE_8	Num
ADATE_9	Num
ADATE_10	Num
ADATE_11	Num
ADATE_12	Num
ADATE_13	Num
ADATE_14	Num
ADATE_15	Num
ADATE_16	Num
ADATE_17	Num
ADATE_18	Num
ADATE_19	Num
ADATE_20	Num
ADATE_21	Num
ADATE_22	Num
ADATE_23	Num
ADATE_24	Num
RFA_2	Char
RFA_3	Char
RFA_4	Char
RFA_5	Char
RFA_6	Char
RFA_7	Char
RFA_8	Char
RFA_9	Char
RFA_10	Char
RFA_11	Char
RFA_12	Char
RFA_13	Char
RFA_14	Char
RFA_15	Char
RFA_16	Char
RFA_17	Char
RFA_18	Char
RFA_19	Char
RFA_20	Char
RFA_21	Char
RFA_22	Char
RFA_23	Char
RFA_24	Char
CARDPROM	Num
MAXADATE	Num
NUMPROM	Num
CARDPM12	Num
NUMPRM12	Num
RDATE_3	Num
RDATE_4	Num
RDATE_5	Num
RDATE_6	Num
RDATE_7	Num

B. KDD Cup Documents

RDATE_8	Num
RDATE_9	Num
RDATE_10	Num
RDATE_11	Num
RDATE_12	Num
RDATE_13	Num
RDATE_14	Num
RDATE_15	Num
RDATE_16	Num
RDATE_17	Num
RDATE_18	Num
RDATE_19	Num
RDATE_20	Num
RDATE_21	Num
RDATE_22	Num
RDATE_23	Num
RDATE_24	Num
RAMNT_3	Num
RAMNT_4	Num
RAMNT_5	Num
RAMNT_6	Num
RAMNT_7	Num
RAMNT_8	Num
RAMNT_9	Num
RAMNT_10	Num
RAMNT_11	Num
RAMNT_12	Num
RAMNT_13	Num
RAMNT_14	Num
RAMNT_15	Num
RAMNT_16	Num
RAMNT_17	Num
RAMNT_18	Num
RAMNT_19	Num
RAMNT_20	Num
RAMNT_21	Num
RAMNT_22	Num
RAMNT_23	Num
RAMNT_24	Num
RAMNTALL	Num
NGIFTALL	Num
CARDGIFT	Num
MINRAMNT	Num
MINRDATE	Num
MAXRAMNT	Num
MAXRDATE	Num
LASTGIFT	Num
LASTDATE	Num
FISTDATE	Num
NEXTDATE	Num
TIMELAG	Num
AVGGIFT	Num
CONTROLN	Num
TARGET_B	Num /* not included in the validation file */
TARGET_D	Num /* not included in the validation file */
HPHONE_D	Num
RFA_2R	Char

B. KDD Cup Documents

RFA_2F	Char
RFA_2A	Char
MDMAUD_R	Char
MDMAUD_F	Char
MDMAUD_A	Char
CLUSTER2	Num
GEOCODE2	Char.

```
+-----+
| SUMMARY STATISTICS (MIN & MAX) |
+-----+
```

Summary statistics are provided for the numeric variables only.

Variable	Learning Data Set		Validation Data Set	
	Minimum	Maximum	Minimum	Maximum
ODATEDW	8306.00	9701.00	8301.00	9701.00
TCODE	0	72002.00	0	39002.00
DOB	0	9710.00	0	9705.00
AGE	1.0000000	98.0000000	1.0000000	98.0000000
NUMCHLD	1.0000000	7.0000000	1.0000000	7.0000000
INCOME	1.0000000	7.0000000	1.0000000	7.0000000
WEALTH1	0	9.0000000	0	9.0000000
HIT	0	241.0000000	0	242.0000000
MBCRAFT	0	6.0000000	0	6.0000000
MBGARDEN	0	4.0000000	0	3.0000000
MBBOOKS	0	9.0000000	0	9.0000000
MBCOLECT	0	6.0000000	0	6.0000000
MAGFAML	0	9.0000000	0	9.0000000
MAGFEM	0	5.0000000	0	4.0000000
MAGMALE	0	4.0000000	0	4.0000000
PUBGARDN	0	5.0000000	0	6.0000000
PUBCULIN	0	6.0000000	0	4.0000000
PUBHLTH	0	9.0000000	0	9.0000000
PUBDOITY	0	8.0000000	0	9.0000000
PUBNEWFN	0	9.0000000	0	9.0000000
PUBPHOTO	0	2.0000000	0	2.0000000
PUBOPP	0	9.0000000	0	9.0000000
MALEMILI	0	99.0000000	0	99.0000000
MALEVET	0	99.0000000	0	99.0000000
VIETVETS	0	99.0000000	0	99.0000000
WWIIVETS	0	99.0000000	0	99.0000000
LOCALGOV	0	99.0000000	0	76.0000000
STATEGOV	0	99.0000000	0	99.0000000
FEDGOV	0	87.0000000	0	99.0000000
WEALTH2	0	9.0000000	0	9.0000000
POP901	0	98701.00	0	100286.00
POP902	0	23766.00	0	21036.00
POP903	0	35403.00	0	35403.00
POP90C1	0	99.0000000	0	99.0000000
POP90C2	0	99.0000000	0	99.0000000
POP90C3	0	99.0000000	0	99.0000000
POP90C4	0	99.0000000	0	99.0000000
POP90C5	0	99.0000000	0	99.0000000

B. KDD Cup Documents

ETH1	0	99.0000000	0	99.0000000
ETH2	0	99.0000000	0	99.0000000
ETH3	0	99.0000000	0	99.0000000
ETH4	0	99.0000000	0	94.0000000
ETH5	0	99.0000000	0	99.0000000
ETH6	0	22.0000000	0	29.0000000
ETH7	0	72.0000000	0	67.0000000
ETH8	0	99.0000000	0	87.0000000
ETH9	0	67.0000000	0	67.0000000
ETH10	0	46.0000000	0	45.0000000
ETH11	0	47.0000000	0	49.0000000
ETH12	0	72.0000000	0	79.0000000
ETH13	0	97.0000000	0	96.0000000
ETH14	0	57.0000000	0	52.0000000
ETH15	0	81.0000000	0	81.0000000
ETH16	0	86.0000000	0	81.0000000
AGE901	0	84.0000000	0	84.0000000
AGE902	0	84.0000000	0	84.0000000
AGE903	0	84.0000000	0	84.0000000
AGE904	0	84.0000000	0	81.0000000
AGE905	0	84.0000000	0	81.0000000
AGE906	0	84.0000000	0	81.0000000
AGE907	0	75.0000000	0	71.0000000
CHIL1	0	99.0000000	0	99.0000000
CHIL2	0	99.0000000	0	99.0000000
CHIL3	0	99.0000000	0	99.0000000
AGEC1	0	99.0000000	0	97.0000000
AGEC2	0	99.0000000	0	99.0000000
AGEC3	0	99.0000000	0	99.0000000
AGEC4	0	99.0000000	0	50.0000000
AGEC5	0	99.0000000	0	99.0000000
AGEC6	0	99.0000000	0	99.0000000
AGEC7	0	99.0000000	0	90.0000000
CHILC1	0	99.0000000	0	99.0000000
CHILC2	0	99.0000000	0	99.0000000
CHILC3	0	99.0000000	0	99.0000000
CHILC4	0	99.0000000	0	99.0000000
CHILC5	0	99.0000000	0	99.0000000
HHAGE1	0	99.0000000	0	99.0000000
HHAGE2	0	99.0000000	0	99.0000000
HHAGE3	0	99.0000000	0	99.0000000
HHN1	0	99.0000000	0	99.0000000
HHN2	0	99.0000000	0	99.0000000
HHN3	0	99.0000000	0	99.0000000
HHN4	0	99.0000000	0	99.0000000
HHN5	0	99.0000000	0	99.0000000
HHN6	0	99.0000000	0	99.0000000
MARR1	0	99.0000000	0	99.0000000
MARR2	0	99.0000000	0	99.0000000
MARR3	0	73.0000000	0	99.0000000
MARR4	0	99.0000000	0	99.0000000
HHP1	0	650.0000000	0	650.0000000
HHP2	0	700.0000000	0	700.0000000
DW1	0	99.0000000	0	99.0000000
DW2	0	99.0000000	0	99.0000000
DW3	0	99.0000000	0	88.0000000
DW4	0	99.0000000	0	99.0000000

B. KDD Cup Documents

DW5	0	99.0000000	0	99.0000000
DW6	0	99.0000000	0	99.0000000
DW7	0	99.0000000	0	99.0000000
DW8	0	99.0000000	0	99.0000000
DW9	0	99.0000000	0	99.0000000
HV1	0	6000.00	0	6000.00
HV2	0	6000.00	0	6000.00
HV3	0	13.0000000	0	13.0000000
HV4	0	13.0000000	0	13.0000000
HU1	0	99.0000000	0	99.0000000
HU2	0	99.0000000	0	99.0000000
HU3	0	99.0000000	0	99.0000000
HU4	0	99.0000000	0	99.0000000
HU5	0	99.0000000	0	99.0000000
HHD1	0	99.0000000	0	99.0000000
HHD2	0	99.0000000	0	99.0000000
HHD3	0	99.0000000	0	99.0000000
HHD4	0	99.0000000	0	99.0000000
HHD5	0	99.0000000	0	99.0000000
HHD6	0	99.0000000	0	99.0000000
HHD7	0	99.0000000	0	99.0000000
HHD8	0	50.0000000	0	31.0000000
HHD9	0	99.0000000	0	99.0000000
HHD10	0	99.0000000	0	99.0000000
HHD11	0	99.0000000	0	99.0000000
HHD12	0	99.0000000	0	99.0000000
ETHC1	0	75.0000000	0	71.0000000
ETHC2	0	99.0000000	0	99.0000000
ETHC3	0	99.0000000	0	99.0000000
ETHC4	0	55.0000000	0	46.0000000
ETHC5	0	99.0000000	0	83.0000000
ETHC6	0	99.0000000	0	80.0000000
HVP1	0	99.0000000	0	99.0000000
HVP2	0	99.0000000	0	99.0000000
HVP3	0	99.0000000	0	99.0000000
HVP4	0	99.0000000	0	99.0000000
HVP5	0	99.0000000	0	99.0000000
HVP6	0	99.0000000	0	99.0000000
HUR1	0	99.0000000	0	99.0000000
HUR2	0	99.0000000	0	99.0000000
RHP1	0	85.0000000	0	85.0000000
RHP2	0	90.0000000	0	90.0000000
RHP3	0	61.0000000	0	61.0000000
RHP4	0	40.0000000	0	40.0000000
HUPA1	0	99.0000000	0	99.0000000
HUPA2	0	99.0000000	0	99.0000000
HUPA3	0	99.0000000	0	99.0000000
HUPA4	0	99.0000000	0	99.0000000
HUPA5	0	99.0000000	0	99.0000000
HUPA6	0	99.0000000	0	99.0000000
HUPA7	0	99.0000000	0	99.0000000
RP1	0	99.0000000	0	99.0000000
RP2	0	99.0000000	0	99.0000000
RP3	0	99.0000000	0	99.0000000
RP4	0	99.0000000	0	99.0000000
MSA	0	9360.00	0	9360.00
ADI	0	651.0000000	0	645.0000000

B. KDD Cup Documents

DMA	0	881.0000000	0	881.0000000
IC1	0	1500.00	0	1500.00
IC2	0	1500.00	0	1500.00
IC3	0	1500.00	0	1394.00
IC4	0	1500.00	0	1500.00
IC5	0	174523.00	0	174523.00
IC6	0	99.0000000	0	99.0000000
IC7	0	99.0000000	0	99.0000000
IC8	0	99.0000000	0	99.0000000
IC9	0	99.0000000	0	99.0000000
IC10	0	99.0000000	0	99.0000000
IC11	0	99.0000000	0	99.0000000
IC12	0	50.0000000	0	57.0000000
IC13	0	61.0000000	0	61.0000000
IC14	0	99.0000000	0	78.0000000
IC15	0	99.0000000	0	99.0000000
IC16	0	99.0000000	0	99.0000000
IC17	0	99.0000000	0	99.0000000
IC18	0	99.0000000	0	99.0000000
IC19	0	99.0000000	0	99.0000000
IC20	0	99.0000000	0	99.0000000
IC21	0	50.0000000	0	99.0000000
IC22	0	99.0000000	0	99.0000000
IC23	0	99.0000000	0	99.0000000
HHAS1	0	99.0000000	0	99.0000000
HHAS2	0	99.0000000	0	99.0000000
HHAS3	0	99.0000000	0	99.0000000
HHAS4	0	99.0000000	0	99.0000000
MC1	0	99.0000000	0	99.0000000
MC2	0	99.0000000	0	99.0000000
MC3	0	99.0000000	0	99.0000000
TPE1	0	99.0000000	0	99.0000000
TPE2	0	99.0000000	0	99.0000000
TPE3	0	99.0000000	0	99.0000000
TPE4	0	99.0000000	0	99.0000000
TPE5	0	71.0000000	0	68.0000000
TPE6	0	47.0000000	0	47.0000000
TPE7	0	25.0000000	0	44.0000000
TPE8	0	99.0000000	0	99.0000000
TPE9	0	99.0000000	0	99.0000000
PEC1	0	99.0000000	0	97.0000000
PEC2	0	99.0000000	0	99.0000000
TPE10	0	90.0000000	0	90.0000000
TPE11	0	76.0000000	0	76.0000000
TPE12	0	99.0000000	0	85.0000000
TPE13	0	99.0000000	0	99.0000000
LFC1	0	99.0000000	0	99.0000000
LFC2	0	99.0000000	0	99.0000000
LFC3	0	99.0000000	0	99.0000000
LFC4	0	99.0000000	0	99.0000000
LFC5	0	99.0000000	0	99.0000000
LFC6	0	99.0000000	0	99.0000000
LFC7	0	99.0000000	0	99.0000000
LFC8	0	99.0000000	0	99.0000000
LFC9	0	99.0000000	0	99.0000000
LFC10	0	99.0000000	0	99.0000000
OCC1	0	99.0000000	0	99.0000000

B. KDD Cup Documents

OCC2	0	99.0000000	0	99.0000000
OCC3	0	99.0000000	0	99.0000000
OCC4	0	99.0000000	0	99.0000000
OCC5	0	99.0000000	0	99.0000000
OCC6	0	43.0000000	0	44.0000000
OCC7	0	55.0000000	0	55.0000000
OCC8	0	99.0000000	0	99.0000000
OCC9	0	99.0000000	0	99.0000000
OCC10	0	99.0000000	0	99.0000000
OCC11	0	99.0000000	0	99.0000000
OCC12	0	99.0000000	0	99.0000000
OCC13	0	99.0000000	0	88.0000000
EIC1	0	99.0000000	0	99.0000000
EIC2	0	65.0000000	0	65.0000000
EIC3	0	99.0000000	0	99.0000000
EIC4	0	99.0000000	0	99.0000000
EIC5	0	99.0000000	0	99.0000000
EIC6	0	64.0000000	0	99.0000000
EIC7	0	99.0000000	0	57.0000000
EIC8	0	99.0000000	0	99.0000000
EIC9	0	99.0000000	0	99.0000000
EIC10	0	99.0000000	0	99.0000000
EIC11	0	99.0000000	0	99.0000000
EIC12	0	67.0000000	0	61.0000000
EIC13	0	99.0000000	0	99.0000000
EIC14	0	99.0000000	0	72.0000000
EIC15	0	99.0000000	0	99.0000000
EIC16	0	99.0000000	0	71.0000000
OEDC1	0	99.0000000	0	99.0000000
OEDC2	0	99.0000000	0	74.0000000
OEDC3	0	99.0000000	0	99.0000000
OEDC4	0	99.0000000	0	99.0000000
OEDC5	0	99.0000000	0	99.0000000
OEDC6	0	99.0000000	0	99.0000000
OEDC7	0	99.0000000	0	99.0000000
EC1	0	170.0000000	0	170.0000000
EC2	0	99.0000000	0	99.0000000
EC3	0	99.0000000	0	99.0000000
EC4	0	99.0000000	0	99.0000000
EC5	0	99.0000000	0	99.0000000
EC6	0	37.0000000	0	68.0000000
EC7	0	99.0000000	0	99.0000000
EC8	0	99.0000000	0	74.0000000
SEC1	0	97.0000000	0	91.0000000
SEC2	0	99.0000000	0	99.0000000
SEC3	0	30.0000000	0	20.0000000
SEC4	0	72.0000000	0	72.0000000
SEC5	0	99.0000000	0	99.0000000
AFC1	0	97.0000000	0	95.0000000
AFC2	0	99.0000000	0	98.0000000
AFC3	0	78.0000000	0	78.0000000
AFC4	0	99.0000000	0	99.0000000
AFC5	0	99.0000000	0	99.0000000
AFC6	0	30.0000000	0	50.0000000
VC1	0	99.0000000	0	99.0000000
VC2	0	99.0000000	0	99.0000000
VC3	0	99.0000000	0	99.0000000

B. KDD Cup Documents

VC4	0	99.0000000	0	99.0000000
ANC1	0	83.0000000	0	74.0000000
ANC2	0	99.0000000	0	73.0000000
ANC3	0	31.0000000	0	41.0000000
ANC4	0	92.0000000	0	99.0000000
ANC5	0	47.0000000	0	48.0000000
ANC6	0	14.0000000	0	23.0000000
ANC7	0	99.0000000	0	57.0000000
ANC8	0	55.0000000	0	99.0000000
ANC9	0	68.0000000	0	57.0000000
ANC10	0	99.0000000	0	74.0000000
ANC11	0	43.0000000	0	74.0000000
ANC12	0	52.0000000	0	38.0000000
ANC13	0	50.0000000	0	50.0000000
ANC14	0	27.0000000	0	33.0000000
ANC15	0	32.0000000	0	47.0000000
P0BC1	0	99.0000000	0	99.0000000
P0BC2	0	99.0000000	0	99.0000000
LSC1	0	99.0000000	0	99.0000000
LSC2	0	99.0000000	0	99.0000000
LSC3	0	99.0000000	0	99.0000000
LSC4	0	99.0000000	0	99.0000000
VOC1	0	99.0000000	0	99.0000000
VOC2	0	99.0000000	0	99.0000000
VOC3	0	99.0000000	0	99.0000000
HC1	0	31.0000000	0	31.0000000
HC2	0	52.0000000	0	52.0000000
HC3	0	99.0000000	0	99.0000000
HC4	0	99.0000000	0	99.0000000
HC5	0	99.0000000	0	99.0000000
HC6	0	99.0000000	0	99.0000000
HC7	0	99.0000000	0	99.0000000
HC8	0	99.0000000	0	99.0000000
HC9	0	90.0000000	0	91.0000000
HC10	0	62.0000000	0	62.0000000
HC11	0	99.0000000	0	99.0000000
HC12	0	99.0000000	0	99.0000000
HC13	0	99.0000000	0	99.0000000
HC14	0	99.0000000	0	99.0000000
HC15	0	30.0000000	0	34.0000000
HC16	0	99.0000000	0	99.0000000
HC17	0	99.0000000	0	99.0000000
HC18	0	99.0000000	0	99.0000000
HC19	0	99.0000000	0	99.0000000
HC20	0	99.0000000	0	99.0000000
HC21	0	99.0000000	0	99.0000000
MHUC1	0	21.0000000	0	21.0000000
MHUC2	0	5.0000000	0	5.0000000
AC1	0	99.0000000	0	52.0000000
AC2	0	99.0000000	0	99.0000000
ADATE_2	9704.00	9706.00	9704.00	9706.00
ADATE_3	9604.00	9606.00	9604.00	9606.00
ADATE_4	9511.00	9609.00	9511.00	9609.00
ADATE_5	9604.00	9604.00	9604.00	9604.00
ADATE_6	9601.00	9603.00	9601.00	9603.00
ADATE_7	9512.00	9602.00	9512.00	9602.00
ADATE_8	9511.00	9605.00	9511.00	9603.00

B. KDD Cup Documents

ADATE_9	9509.00	9511.00	9509.00	9511.00
ADATE_10	9510.00	9511.00	9510.00	9511.00
ADATE_11	9508.00	9511.00	9508.00	9511.00
ADATE_12	9507.00	9510.00	9507.00	9510.00
ADATE_13	9502.00	9507.00	9502.00	9507.00
ADATE_14	9504.00	9506.00	9504.00	9506.00
ADATE_15	9504.00	9504.00	9504.00	9504.00
ADATE_16	9502.00	9504.00	9502.00	9504.00
ADATE_17	9501.00	9503.00	9501.00	9503.00
ADATE_18	9409.00	9508.00	9409.00	9508.00
ADATE_19	9409.00	9411.00	9409.00	9411.00
ADATE_20	9411.00	9412.00	9411.00	9412.00
ADATE_21	9409.00	9410.00	9409.00	9410.00
ADATE_22	9408.00	9506.00	9408.00	9506.00
ADATE_23	9312.00	9407.00	9312.00	9407.00
ADATE_24	9405.00	9406.00	9405.00	9406.00
CARDPROM	1.0000000	61.0000000	0	62.0000000
MAXADATE	9608.00	9702.00	9607.00	9702.00
NUMPROM	4.0000000	195.0000000	4.0000000	189.0000000
CARDPM12	0	19.0000000	0	21.0000000
NUMPRM12	1.0000000	78.0000000	1.0000000	76.0000000
RDATE_3	9605.00	9806.00	9309.00	9806.00
RDATE_4	9510.00	9804.00	9509.00	9805.00
RDATE_5	9604.00	9803.00	9604.00	9805.00
RDATE_6	9510.00	9805.00	9511.00	9806.00
RDATE_7	9512.00	9610.00	9511.00	9701.00
RDATE_8	9511.00	9806.00	9512.00	9806.00
RDATE_9	9509.00	9609.00	9509.00	9603.00
RDATE_10	9510.00	9806.00	9511.00	9804.00
RDATE_11	9509.00	9805.00	9509.00	9606.00
RDATE_12	9509.00	9806.00	9509.00	9804.00
RDATE_13	9502.00	9603.00	9502.00	9803.00
RDATE_14	9406.00	9603.00	9505.00	9603.00
RDATE_15	9412.00	9603.00	9412.00	9603.00
RDATE_16	9411.00	9805.00	9410.00	9603.00
RDATE_17	9502.00	9512.00	9502.00	9512.00
RDATE_18	9412.00	9601.00	9407.00	9602.00
RDATE_19	9409.00	9509.00	9409.00	9509.00
RDATE_20	9411.00	9508.00	9411.00	9508.00
RDATE_21	9409.00	9508.00	9409.00	9508.00
RDATE_22	9409.00	9510.00	9409.00	9508.00
RDATE_23	9309.00	9507.00	9309.00	9507.00
RDATE_24	9309.00	9504.00	9309.00	9504.00
RAMNT_3	2.0000000	50.0000000	2.0000000	200.0000000
RAMNT_4	1.0000000	100.0000000	1.0000000	100.0000000
RAMNT_5	4.0000000	50.0000000	5.0000000	30.0000000
RAMNT_6	1.0000000	100.0000000	1.0000000	100.0000000
RAMNT_7	1.0000000	250.0000000	1.0000000	203.0000000
RAMNT_8	1.0000000	500.0000000	0.3200000	3713.31
RAMNT_9	1.0000000	1000.00	1.0000000	300.0000000
RAMNT_10	0.3000000	500.0000000	1.0000000	10000.00
RAMNT_11	1.0000000	300.0000000	1.0000000	1000.00
RAMNT_12	1.0000000	300.0000000	1.0000000	500.0000000
RAMNT_13	0.1000000	500.0000000	1.0000000	300.0000000
RAMNT_14	1.0000000	200.0000000	1.0000000	600.0000000
RAMNT_15	1.0000000	300.0000000	1.0000000	500.0000000
RAMNT_16	0.5000000	500.0000000	0.5000000	205.0000000

B. KDD Cup Documents

RAMNT_17	1.0000000	500.0000000	1.0000000	500.0000000
RAMNT_18	1.0000000	1000.00	0.3200000	300.0000000
RAMNT_19	1.0000000	970.0000000	1.0000000	250.0000000
RAMNT_20	0.5000000	250.0000000	1.0000000	200.0000000
RAMNT_21	1.0000000	300.0000000	1.0000000	1000.00
RAMNT_22	0.2900000	300.0000000	1.0000000	500.0000000
RAMNT_23	0.3000000	200.0000000	1.0000000	300.0000000
RAMNT_24	1.0000000	225.0000000	0.5000000	250.0000000
RAMNTALL	13.0000000	9485.00	13.0000000	10253.00
NGIFTALL	1.0000000	237.0000000	1.0000000	126.0000000
CARDGIFT	0	41.0000000	0	45.0000000
MINRAMNT	0	1000.00	0	436.0000000
MINRDATE	7506.00	9702.00	8010.00	9702.00
MAXRAMNT	5.0000000	5000.00	5.0000000	10000.00
MAXRDATE	7510.00	9702.00	8011.00	9702.00
LASTGIFT	0	1000.00	0	10000.00
LASTDATE	9503.00	9702.00	9503.00	9702.00
FISTDATE	0	9603.00	0	9603.00
NEXTDATE	7211.00	9702.00	7312.00	9702.00
TIMELAG	0	1088.00	0	1060.00
AVGGIFT	1.2857143	1000.00	1.5789474	650.0000000
CONTROLN	1.0000000	191779.00	3.0000000	191776.00
TARGET_B	0	1.0000000	0	1.0000000
TARGET_D	0	200.0000000	0	500.0000000
HPHONE_D	0	1.0000000	0	1.0000000
CLUSTER2	1.0000000	62.0000000	1.0000000	62.0000000

```
+-----+
| DATA (PRE)PROCESSING |
+-----+
```

General

- o The field CONTROLN is a unique record identifier (an index) and should not be used in modeling
- o Response flag (field name: TARGET_B) indicates whether or not the lapsed donor responded to the campaign. THIS FIELD SHOULD NOT BE USED DURING MODEL BUILDING.
- o Blanks in string or character variables correspond to missing values. Periods and/or blanks in the numeric variables correspond to missing values.

Data preprocessing tasks include the following:

Noisy Data

Some of the fields in the analysis file may contain data entry and/or formatting errors. You are expected to clean these fields (without excluding the records.)

Records and Fields with Missing and Sparse Data

B. KDD Cup Documents

Discovery methods vary in the way they treat the missing values. While some simply disregard missing values or omit the corresponding records, others infer missing values from known values, or treat missing data as a special value to be included additionally in the attribute domain.

For the purposes of KDD-CUP-98 the records and/or fields should not be omitted from analysis because they contain missing data. Instead, the missing data should be inferred from known values (e.g., mean, median, mode, a modeled value, or any other way supported by your tool.) One exception to this rule is the attributes containing 99.5 percent or more missings. You are expected to omit these attributes from the analysis.

You are also expected to drop attributes with 'sparse' distributions. Sparse data occur when the events actually represented in given data make only a very small subset of the event space.

Fields Containing Constants

Fields containing a constant value (i.e., there is only one value for all the records) should be dropped from the analysis. Attributes containing missing and one valid level (e.g., 'Y') are not considered as constants and should be included in the analysis.

Time Frame and Date Fields

This mailing was mailed to a total of 3.5 million PVA donors who were on the PVA database as of June 1997. All information contained in the analysis dataset reflects the donor status prior to 6/97 (except the gift receipt dates, which will follow the promotion dates.) This date could be used as the "end date" or "rerefence date" in the calculation of "number of months since" variables.

ATTRIBUTE TYPE

See the data dictionary to determine the attribute types.

+-----+	+-----+
KDD-CUP-98 Program Committee	
+-----+	+-----+

- o Vasant Dhar, New York University, New York, NY
- o Tom Fawcett, Bell Atlantic, New York, NY
- o Georges Grinstein, University of Massachusetts, Lowell, MA
- o Ismail Parsa, Epsilon, Burlington, MA
- o Gregory Piatetsky-Shapiro, Knowledge Stream Partners, Boston, MA
- o Foster Provost, Bell Atlantic, New York, NY
- o Kyuseok Shim, Bell Laboratories, Murray Hill, NJ

B. KDD Cup Documents

```
+-----+  
| TERMINOLOGY-GLOSSARY |  
+-----+
```

[GLOSSARY]

For more information on the terminology used throughout this documentation, refer to the questionnaire documentation (file name: cup98QUE.txt.)

- o attribute = field = variable = feature
- o responders = targets
- o non-responders = non-targets
- o output = target = dependent variable
- o inputs = independent variables
- o analysis file = analysis sample = combined learning and validation files

```
=====  
EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL
```

INFORMATION LISTED BELOW IS AVAILABLE UNDER THE TERMS OF THE
CONFIDENTIALITY AGREEMENT

```
=====  
EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL
```

B.2. Data Set Dictionary

```
=====  
EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL
```

INFORMATION LISTED BELOW IS AVAILABLE UNDER THE TERMS OF THE
CONFIDENTIALITY AGREEMENT

```
=====  
EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL      EPSILON CONFIDENTIAL
```

```
+-----+  
|            PARALYZED VETERANS OF AMERICA (PVA)        |  
|            DATA DICTIONARY TO ACCOMPANY                |  
|                                                               |  
|            KDD-CUP-98                                        |  
|                                                               |  
|            The Second International Knowledge Discovery and        |  
|            Data Mining Tools Competition                 |  
|                                                               |  
|            Held in Conjunction with KDD-98                |  
|                                                               |  
|            The Fourth International Conference on Knowledge        |  
|            Discovery and Data Mining                        |  
|            [www.kdnuggets.com] or                            |  
|            [www-aig.jpl.nasa.gov/kdd98] or                |  
|            [www.aaai.org/Conferences/KDD/1998]                |  
+-----+
```

B. KDD Cup Documents

```
|           Sponsored by the
|           |
|           American Association for Artificial Intelligence (AAAI)
|           Epsilon Data Mining Laboratory
|           Paralyzed Veterans of America (PVA)
+-----+
|   Created:    7/20/98
|   Last update: 7/20/98
|   file name:   cup98DIC.txt
+-----+
```

Variable	Description
ODATEDW	Origin Date. Date of donor's first gift to PVA YYMM format (Year/Month).
OSOURCE	Origin Source - (Only 1st 3 bytes are used) - Defaulted to 00000 for conversion - Code indicating which mailing list the donor was originally acquired from - A nominal or symbolic field.
TCODE	Donor title code 000 = _ 001 = MR. 001001 = MESSRS. 001002 = MR. & MRS. 002 = MRS. 002002 = MESDAMES 003 = MISS 003003 = MISSES 004 = DR. 004002 = DR. & MRS. 004004 = DOCTORS 005 = MADAME 006 = SERGEANT 009 = RABBI 010 = PROFESSOR 010002 = PROFESSOR & MRS. 010010 = PROFESSORS 011 = ADMIRAL 011002 = ADMIRAL & MRS. 012 = GENERAL 012002 = GENERAL & MRS. 013 = COLONEL 013002 = COLONEL & MRS. 014 = CAPTAIN 014002 = CAPTAIN & MRS. 015 = COMMANDER 015002 = COMMANDER & MRS. 016 = DEAN 017 = JUDGE 017002 = JUDGE & MRS.

B. KDD Cup Documents

018 = MAJOR
018002 = MAJOR & MRS.
019 = SENATOR
020 = GOVERNOR
021002 = SERGEANT & MRS.
022002 = COLNEL & MRS.
024 = LIEUTENANT
026 = MONSIGNOR
027 = REVEREND
028 = MS.
028028 = MSS.
029 = BISHOP
031 = AMBASSADOR
031002 = AMBASSADOR & MRS.
033 = CANTOR
036 = BROTHER
037 = SIR
038 = COMMODORE
040 = FATHER
042 = SISTER
043 = PRESIDENT
044 = MASTER
046 = MOTHER
047 = CHAPLAIN
048 = CORPORAL
050 = ELDER
056 = MAYOR
059002 = LIEUTENANT & MRS.
062 = LORD
063 = CARDINAL
064 = FRIEND
065 = FRIENDS
068 = ARCHDEACON
069 = CANON
070 = BISHOP
072002 = REVEREND & MRS.
073 = PASTOR
075 = ARCHBISHOP
085 = SPECIALIST
087 = PRIVATE
089 = SEAMAN
090 = AIRMAN
091 = JUSTICE
092 = MR. JUSTICE
100 = M.
103 = MLLE.
104 = CHANCELLOR
106 = REPRESENTATIVE
107 = SECRETARY
108 = LT. GOVERNOR
109 = LIC.
111 = SA.
114 = DA.
116 = SR.
117 = SRA.
118 = SRTA.
120 = YOUR MAJESTY

B. KDD Cup Documents

122	= HIS HIGHNESS
123	= HER HIGHNESS
124	= COUNT
125	= LADY
126	= PRINCE
127	= PRINCESS
128	= CHIEF
129	= BARON
130	= SHEIK
131	= PRINCE AND PRINCESS
132	= YOUR IMPERIAL MAJEST
135	= M. ET MME.
210	= PROF.
STATE	State abbreviation (a nominal/symbolic field)
ZIP	Zipcode (a nominal/symbolic field)
MAILCODE	Mail Code " " = Address is OK B = Bad Address
PVASTATE	EPVA State or PVA State Indicates whether the donor lives in a state served by the organization's EPVA chapter P = PVA State E = EPVA State (Northeastern US)
DOB	Date of birth (YYMM, Year/Month format.)
NOEXCH	Do Not Exchange Flag (For list rental) _ = can be exchanged X = do not exchange
RECINHSE	In House File Flag _ = Not an In House Record X = Donor has given to PVA's In House program
RECP3	P3 File Flag _ = Not a P3 Record X = Donor has given to PVA's P3 program
RECPVG	Planned Giving File Flag _ = Not a Planned Giving Record X = Planned Giving Record
RECSWEEP	Sweepstakes file flag _ = Not a Sweepstakes Record X = Sweepstakes Record
MDMAUD	The Major Donor Matrix code The codes describe frequency and amount of giving for donors who have given a \$100+ gift at any time in their giving history. An RFA (recency/frequency/monetary) field.
	The (current) concatenated version is a nominal or symbolic field. The individual bytes could separately be used as fields and refer to the following:

B. KDD Cup Documents

First byte: Recency of Giving

C=Current Donor
L=Lapsed Donor
I=Inactive Donor
D=Dormant Donor

2nd byte: Frequency of Giving

1=One gift in the period of recency
2=Two-Four gifts in the period of recency
5=Five+ gifts in the period of recency

3rd byte: Amount of Giving

L=Less than \$100(Low Dollar)
C=\$100-499(Core)
M=\$500-999(Major)
T=\$1,000+(Top)

4th byte: Blank/meaningless/filler

'X' indicates that the donor is not a major donor.

For more information regarding the RFA codes, see
the promotion history field definitions.

DOMAIN

DOMAIN/Cluster code. A nominal or symbolic field.
could be broken down by bytes as explained below.

1st byte = Urbanicity level of the donor's neighborhood

U=Urban
C=City
S=Suburban
T=Town
R=Rural

2nd byte = Socio-Economic status of the neighborhood

1 = Highest SES
2 = Average SES
3 = Lowest SES (except for Urban communities, where
1 = Highest SES, 2 = Above average SES,
3 = Below average SES, 4 = Lowest SES.)

CLUSTER

CLUSTER

Code indicating which cluster group the donor falls into.
Each cluster is unique in terms of socio-economic status,
urbanicity, ethnicity and a variety of other demographic
characteristics. A nominal or symbolic field.

AGE

Overlay Age
0 = missing

AGEFLAG

Age Flag
E = Exact
I = Inferred from Date of Birth Field

HOMEOWNR

Home Owner Flag
H = Home owner
U = Unknown

B. KDD Cup Documents

CHILD03	Presence of Children age 0-3 B = Both, F = Female, M = Male
CHILD07	Presence of Childern age 4-7
CHILD12	Presence of Childern age 8-12
CHILD18	Presence of Childern age 13-18
NUMCHLD	NUMBER OF CHILDREN
INCOME	HOUSEHOLD INCOME
GENDER	Gender M = Male F = Female U = Unknown J = Joint Account, unknown gender
WEALTH1	Wealth Rating
HIT	MOR Flag # HIT (Mail Order Response) Indicates total number of known times the donor has responded to a mail order offer other than PVA's.

The following variables indicate the number of known times the donor has responded to other types of mail order offers.	
MBCRAFT	Buy Craft Hobby
MBGARDEN	Buy Gardening
MBBOOKS	Buy Books
MBCOLECT	Buy Collectables
MAGFAML	Buy General Family Mags
MAGFEM	Buy Female Mags
MAGMALE	Buy Sports Mags
PUBGARDN	Gardening Pubs
PUBCULIN	Culinary Pubs
PUBHLTH	Health Pubs
PUBDOITY	Do It Yourself Pubs
PUBNEWFN	News / Finance Pubs
PUBPHOTO	Photography Pubs
PUBOPP	Opportunity Seekers Pubs

DATASRCE	Source of Overlay Data Indicates which third-party data source the donor matched against 1 = MetroMail 2 = Polk 3 = Both
MALEMILI	% Males active in the Military
MALEVET	% Males Veterans
VIETVETS	% Vietnam Vets
WWIIVETS	% WWII Vets
LOCALGOV	% Employed by Local Gov
STATEGOV	% Employed by State Gov
FEDGOV	% Employed by Fed Gov

B. KDD Cup Documents

SOLP3	SOLICIT LIMITATION CODE P3 = can be mailed (Default) 00 = Do Not Solicit or Mail 01 = one solicitation per year 02 = two solicitations per year 03 = three solicitations per year 04 = four solicitations per year 05 = five solicitations per year 06 = six solicitations per year 12 = twelve solicitations per year
SOLIH	SOLICITATION LIMIT CODE IN HOUSE = can be mailed (Default) 00 = Do Not Solicit 01 = one solicitation per year 02 = two solicitations per year 03 = three solicitations per year 04 = four solicitations per year 05 = five solicitations per year 06 = six solicitations per year 12 = twelve solicitations per year
MAJOR	Major (\$\$) Donor Flag _ = Not a Major Donor X = Major Donor
WEALTH2	Wealth Rating Wealth rating uses median family income and population statistics from each area to index relative wealth within each state The segments are denoted 0-9, with 9 being the highest income group and zero being the lowest. Each rating has a different meaning within each state.
GEOCODE	Geo Cluster Code indicating the level geography at which a record matches the census data. A nominal or symbolic field. Blank=No code has been assigned or did not match at any level.
----- The following variables reflect donor interests, as collected from third-party data sources	
COLLECT1	COLLECTABLE (Y/N)
VETERANS	VETERANS (Y/N)
BIBLE	BIBLE READING (Y/N)
CATLG	SHOP BY CATALOG (Y/N)
HOMEE	WORK FROM HOME (Y/N)
PETS	HOUSEHOLD PETS (Y/N)
CDPLAY	CD PLAYER OWNERS (Y/N)
STEREO	STEREO/RECORDS/TAPES/CD (Y/N)
PCOWNERS	HOME PC OWNERS/USERS
PHOTO	PHOTOGRAPHY (Y/N)
CRAFTS	CRAFTS (Y/N)

B. KDD Cup Documents

FISHER	FISHING (Y/N)
GARDENIN	GARDENING (Y/N)
BOATS	POWER BOATING (Y/N)
WALKER	WALK FOR HEALTH (Y/N)
KIDSTUFF	BUYS CHILDREN'S PRODUCTS (Y/N)
CARDS	STATIONARY/CARDS BUYER (Y/N)
PLATES	PLATE COLLECTOR (Y/N)
 LIFESRC	 LIFE STYLE DATA SOURCE
	Indicates source of the lifestyle variables listed above
	1 = MATCHED ON METRO MAIL ONLY
	2 = MATCHED ON POLK ONLY
	3 = MATCHED BOTH MM AND POLK

PEPSTRFL	Indicates PEP Star RFA Status blank = Not considered to be a PEP Star 'X' = Has PEP Star RFA Status

The following variables reflect characteristics of the donors neighborhood, as collected from the 1990 US Census.	
POP901	Number of Persons
POP902	Number of Families
POP903	Number of Households
POP90C1	Percent Population in Urbanized Area
POP90C2	Percent Population Outside Urbanized Area
POP90C3	Percent Population Inside Rural Area
POP90C4	Percent Male
POP90C5	Percent Female
ETH1	Percent White
ETH2	Percent Black
ETH3	Percent Native American
ETH4	Percent Pacific Islander/Asian
ETH5	Percent Hispanic
ETH6	Percent Asian Indian
ETH7	Percent Japanese
ETH8	Percent Chinese
ETH9	Percent Philipino
ETH10	Percent Korean
ETH11	Percent Vietnamese
ETH12	Percent Hawaiian
ETH13	Percent Mexican
ETH14	Percent Puerto Rican
ETH15	Percent Cuban
ETH16	Percent Other Hispanic
AGE901	Median Age of Population
AGE902	Median Age of Adults 18 or Older
AGE903	Median Age of Adults 25 or Older
AGE904	Average Age of Population
AGE905	Average Age of Adults >= 18
AGE906	Average Age of Adults >= 25

B. KDD Cup Documents

AGE907	Percent Population Under Age 18
CHIL1	Percent Children Under Age 7
CHIL2	Percent Children Age 7 - 13
CHIL3	Percent Children Age 14-17
AGEC1	Percent Adults Age18-24
AGEC2	Percent Adults Age 25-34
AGEC3	Percent Adults Age 35-44
AGEC4	Percent Adults Age 45-54
AGEC5	Percent Adults Age 55-64
AGEC6	Percent Adults Age 65-74
AGEC7	Percent Adults Age >= 75
CHILC1	Percent Children Age <=2
CHILC2	Percent Children Age 3-5
CHILC3	Percent Children Age 6-11
CHILC4	Percent Children Age 12-15
CHILC5	Percent Children Age 16-18
HHAGE1	Percent Households w/ Person 65+
HHAGE2	Percent Households w/ Person 65+ Living Alone
HHAGE3	Percent Households Headed by an Elderly Person Age 65+
HHN1	Percent 1 Person Households
HHN2	Percent 2 Person Households
HHN3	Percent 3 or More Person Households
HHN4	Percent 4 or More Person Households
HHN5	Percent 5 or More Person Households
HHN6	Percent 6 Person Households
MARR1	Percent Married
MARR2	Percent Separated or Divorced
MARR3	Percent Widowed
MARR4	Percent Never Married
HHP1	Median Person Per Household
HHP2	Average Person Per Household
DW1	Percent Single Unit Structure
DW2	Percent Detached Single Unit Structure
DW3	Percent Duplex Structure
DW4	Percent Multi (2+) Unit Structures
DW5	Percent 3+ Unit Structures
DW6	Percent Housing Units in 5+ Unit Structure
DW7	Percent Group Quarters
DW8	Percent Institutional Group Quarters
DW9	Non-Institutional Group Quarters
HV1	Median Home Value in hundreds
HV2	Average Home Value in hundreds
HV3	Median Contract Rent in hundreds
HV4	Average Contract Rent in hundreds
HU1	Percent Owner Occupied Housing Units
HU2	Percent Renter Occupied Housing Units
HU3	Percent Occupied Housing Units
HU4	Percent Vacant Housing Units
HU5	Percent Seasonal/Recreational Vacant Units
HHD1	Percent Households w/ Related Children
HHD2	Percent Households w/ Families
HHD3	Percent Married Couple Families
HHD4	Percent Married Couples w/ Related Children
HHD5	Percent Persons in Family Household
HHD6	Percent Persons in Non-Family Household
HHD7	Percent Single Parent Households
HHD8	Percent Male Householder w/ Child

B. KDD Cup Documents

HHD9	Percent Female Householder w/ Child
HHD10	Percent Single Male Householder
HHD11	Percent Single Female Householder
HHD12	Percent Households w/ Non-Family Living Arrangements
ETHC1	Percent White < Age 15
ETHC2	Percent White Age 15 - 59
ETHC3	Percent White Age 60+
ETHC4	Percent Black < Age 15
ETHC5	Percent Black Age 15 - 59
ETHC6	Percent Black Age 60+
HVP1	Percent Home Value >= \$200,000
HVP2	Percent Home Value >= \$150,000
HVP3	Percent Home Value >= \$100,000
HVP4	Percent Home Value >= \$75,000
HVP5	Percent Home Value >= \$50,000
HVP6	Percent Home Value >= \$300,000
HUR1	\$ 1 or 2 Room Housing Units
HUR2	Percent >= 6 Room Housing Units
RHP1	Median Number of Rooms per Housing Unit
RHP2	Average Number of Rooms per Housing Unit
RHP3	Median Number of Persons per Housing Unit
RHP4	Average Number of Persons per Room
HUPA1	Percent Housing Units w/ 2 thru 9 Units at the Address
HUPA2	Percent Housing Units w/ >= 10 Units at the Address
HUPA3	Percent Mobile Homes or Trailers
HUPA4	Percent Renter Occupied Single Unit Structure
HUPA5	Percent Renter Occupied, 2 - 4 Units
HUPA6	Percent Renter Occupied, 5+ Units
HUPA7	Percent Renter Occupied Mobile Homes or Trailers
RP1	Percent Renters Paying >= \$500 per Month
RP2	Percent Renters Paying >= \$400 per Month
RP3	Percent Renters Paying >= \$300 per Month
RP4	Percent Renters Paying >= \$200 per Month
MSA	MSA Code
ADI	ADI Code
DMA	DMA Code
IC1	Median Household Income in hundreds
IC2	Median Family Income in hundreds
IC3	Average Household Income in hundreds
IC4	Average Family Income in hundreds
IC5	Per Capita Income
IC6	Percent Households w/ Income < \$15,000
IC7	Percent Households w/ Income \$15,000 - \$24,999
IC8	Percent Households w/ Income \$25,000 - \$34,999
IC9	Percent Households w/ Income \$35,000 - \$49,999
IC10	Percent Households w/ Income \$50,000 - \$74,999
IC11	Percent Households w/ Income \$75,000 - \$99,999
IC12	Percent Households w/ Income \$100,000 - \$124,999
IC13	Percent Households w/ Income \$125,000 - \$149,999
IC14	Percent Households w/ Income >= \$150,000
IC15	Percent Families w/ Income < \$15,000
IC16	Percent Families w/ Income \$15,000 - \$24,999
IC17	Percent Families w/ Income \$25,000 - \$34,999
IC18	Percent Families w/ Income \$35,000 - \$49,999
IC19	Percent Families w/ Income \$50,000 - \$74,999
IC20	Percent Families w/ Income \$75,000 - \$99,999
IC21	Percent Families w/ Income \$100,000 - \$124,999

B. KDD Cup Documents

IC22	Percent Families w/ Income \$125,000 - \$149,999
IC23	Percent Families w/ Income >= \$150,000
HHAS1	Percent Households on Social Security
HHAS2	Percent Households on Public Assistance
HHAS3	Percent Households w/ Interest, Rental or Dividend Income
HHAS4	Percent Persons Below Poverty Level
MC1	Percent Persons Move in Since 1985
MC2	Percent Persons in Same House in 1985
MC3	Percent Persons in Different State/Country in 1985
TPE1	Percent Driving to Work Alone Car/Truck/Van
TPE2	Percent Carpooling Car/Truck/Van)
TPE3	Percent Using Public Transportation
TPE4	Percent Using Bus/Trolley
TPE5	Percent Using Railways
TPE6	Percent Using Taxi/Ferry
TPE7	Percent Using Motorcycles
TPE8	Percent Using Other Transportation
TPE9	Percent Working at Home/No Transportation
PEC1	Percent Working Outside State of Residence
PEC2	Percent Working Outside County of Residence in State
TPE10	Median Travel Time to Work in minutes
TPE11	Mean Travel Time to Work in minutes
TPE12	Percent Traveling 60+ Minutes to Work
TPE13	Percent Traveling 15 - 59 Minutes to Work
LFC1	Percent Adults in Labor Force
LFC2	Percent Adult Males in Labor Force
LFC3	Percent Females in Labor Force
LFC4	Percent Adult Males Employed
LFC5	Percent Adult Females Employed
LFC6	Percent Mothers Employed Married and Single
LFC7	Percent 2 Parent Earner Families
LFC8	Percent Single Mother w/ Child in Labor Force
LFC9	Percent Single Father w/ Child in Labor Force
LFC10	Percent Families w/ Child w/ no Workers
OCC1	Percent Professional
OCC2	Percent Managerial
OCC3	Percent Technical
OCC4	Percent Sales
OCC5	Percent Clerical/Administrative Support
OCC6	Percent Private Household Service Occ.
OCC7	Percent Protective Service Occ.
OCC8	Percent Other Service Occ.
OCC9	Percent Farmers
OCC10	Percent Craftsmen, Precision, Repair
OCC11	Percent Operatives, Machine
OCC12	Percent Transportation
OCC13	Percent Laborers, Handlers, Helpers
EIC1	Percent Employed in Agriculture
EIC2	Percent Employed in Mining
EIC3	Percent Employed in Construction
EIC4	Percent Employed in Manufacturing
EIC5	Percent Employed in Transportation
EIC6	Percent Employed in Communications
EIC7	Percent Employed in Wholesale Trade
EIC8	Percent Employed in Retail Industry
EIC9	Percent Employed in Finance, Insurance, Real Estate
EIC10	Percent Employed in Business and Repair

B. KDD Cup Documents

EIC11	Percent Employed in Personnal Services
EIC12	Percent Employed in Entertainment and Recreation
EIC13	Percent Employed in Health Services
EIC14	Percent Employed in Educational Services
EIC15	Percent Employed in Other Professional Services
EIC16	Percent Employed in Public Administration
OEDC1	Percent Employed by Local Government
OEDC2	Percent Employed by State Government
OEDC3	Percent Employed by Federal Government
OEDC4	Percent Self Employed
OEDC5	Percent Private Profit Wage or Salaried Worker
OEDC6	Percent Private Non-Profit Wage or Salaried Worker
OEDC7	Percent Unpaid Family Workers
EC1	Median Years of School Completed by Adults 25+
EC2	Percent Adults 25+ Grades 0-8
EC3	Percent Adults 25+ w/ some High School
EC4	Percent Adults 25+ Completed High School or Equivalency
EC5	Percent Adults 25+ w/ some College
EC6	Percent Adults 25+ w/ Associates Degree
EC7	Percent Adults 25+ w/ Bachelors Degree
EC8	Percent Adults 25+ Graduate Degree
SEC1	Percent Persons Enrolled in Private Schools
SEC2	Percent Persons Enrolled in Public Schools
SEC3	Percent Persons Enrolled in Preschool
SEC4	Percent Persons Enrolled in Elementary or High School
SEC5	Percent Persons in College
AFC1	Percent Adults in Active Military Service
AFC2	Percent Males in Active Military Service
AFC3	Percent Females in Active Military Service
AFC4	Percent Adult Veterans Age 16+
AFC5	Percent Male Veterans Age 16+
AFC6	Percent Female Veterans Age 16+
VC1	Percent Vietnam Veterans Age 16+
VC2	Percent Korean Veterans Age 16+
VC3	Percent WW2 Veterans Age 16+
VC4	Percent Veterans Serving After May 1975 Only
ANC1	Percent Dutch Ancestry
ANC2	Percent English Ancestry
ANC3	Percent French Ancestry
ANC4	Percent German Ancestry
ANC5	Percent Greek Ancestry
ANC6	Percent Hungarian Ancestry
ANC7	Percent Irish Ancestry
ANC8	Percent Italian Ancestry
ANC9	Percent Norwegian Ancestry
ANC10	Percent Polish Ancestry
ANC11	Percent Portuguese Ancestry
ANC12	Percent Russian Ancestry
ANC13	Percent Scottish Ancestry
ANC14	Percent Swedish Ancestry
ANC15	Percent Ukrainian Ancestry
POBC1	Percent Foreign Born
POBC2	Percent Born in State of Residence
LSC1	Percent English Only Speaking
LSC2	Percent Spanish Speaking
LSC3	Percent Asian Speaking
LSC4	Percent Other Language Speaking

B. KDD Cup Documents

VOC1	Percent Households w/ 1+ Vehicles
VOC2	Percent Households w/ 2+ Vehicles
VOC3	Percent Households w/ 3+ Vehicles
HC1	Percent Median Length of Residence
HC2	Percent Median Age of Occupied Dwellings in years
HC3	Percent Owner Occupied Structures Built Since 1989
HC4	Percent Owner Occupied Structures Built Since 1985
HC5	Percent Owner Occupied Structures Built Since 1980
HC6	Percent Owner Occupied Structures Built Since 1970
HC7	Percent Owner Occupied Structures Built Since 1960
HC8	Percent Owner Occupied Structures Built Prior to 1960
HC9	Percent Owner Occupied Condominiums
HC10	Percent Renter Occupied Condominiums
HC11	Percent Occupied Housing Units Heated by Utility Gas
HC12	Percent Occupied Housing Units Heated by Bottled, Tank or LP
HC13	Percent Occupied Housing Units Heated by Electricity
HC14	Percent Occupied Housing Units Heated by Fuel Oil
HC15	Percent Occupied Housing Units Heated by Solar Energy
HC16	Percent Occupied Housing Units Heated by Coal, Wood, Other
HC17	Percent Housing Units w/ Public Water Source
HC18	Percent Housing Units w/ Well Water Source
HC19	Percent Housing Units w/ Public Sewer Source
HC20	Percent Housing Units w/ Complete Plumbing Facilities
HC21	Percent Housing Units w/ Telephones
MHUC1	Median Homeowner Cost w/ Mortgage per Month dollars
MHUC2	Median Homeowner Cost w/out Mortgage per Month dollars
AC1	Percent Adults Age 55-59
AC2	Percent Adults Age 60-64

The fields listed below are from the promotion history file.

PROMOTION CODES:

The following lists the promotion codes and their respective field names (where XXXX refers to ADATE, RFA, RDATE and RAMNT.)

```
'97NK' ==> xxxx_2 (mailing was used to construct  
                     the target fields)  
'96NK' ==> xxxx_3  
'96TK' ==> xxxx_4  
'96SK' ==> xxxx_5  
'96LL' ==> xxxx_6  
'96G1' ==> xxxx_7  
'96GK' ==> xxxx_8  
'96CC' ==> xxxx_9  
'96WL' ==> xxxx_10  
'96X1' ==> xxxx_11  
'96XK' ==> xxxx_12  
'95FS' ==> xxxx_13  
'95NK' ==> xxxx_14  
'95TK' ==> xxxx_15  
'95LL' ==> xxxx_16  
'95G1' ==> xxxx_17
```

B. KDD Cup Documents

```
'95GK' ==> xxxx_18
'95CC' ==> xxxx_19
'95WL' ==> xxxx_20
'95X1' ==> xxxx_21
'95XK' ==> xxxx_22
'94FS' ==> xxxx_23
'94NK' ==> xxxx_24
```

1st 2 bytes of the code refers to the year of the mailing while 3rd and 4th bytes refer to the following promotion codes/types:

```
LL mailings had labels only
WL mailings had labels only
CC mailings are calendars with stickers but do
    not have labels
FS mailings are blank cards that fold into
    thirds with labels
NK mailings are blank cards with labels
SK mailings are blank cards with labels
TK mailings have thank you printed on the
    outside with labels
GK mailings are general greeting cards (an
    assortment of birthday, sympathy, blank, & get
    well) with labels
XK mailings are Christmas cards with labels
X1 mailings have labels and a notepad
G1 mailings have labels and a notepad
```

This information could certainly be used to calculate several summary variables that count the number of occurrences of various types of promotions received in the most recent 12-36 months, etc.

RFA (RECENCY/FREQUENCY/AMOUNT)

The RFA (recency/frequency/amount) status of the donors (as of the promotion dates) is included in the RFA fields.

The (current) concatenated version is a nominal or symbolic field. The individual bytes could separately be used as fields and refer to the following:

First Byte of code is concerned with RECENCY based on Date of the last Gift

F=FIRST TIME DONOR Anyone who has made their first donation in the last 6 months and has made just one donation.

N=NEW DONOR Anyone who has made their first donation in the last 12 months and is not a First time donor. This is everyone who made their first donation 7-12 months ago, or

B. KDD Cup Documents

people who made their first donation between 0-6 months ago and have made 2 or more donations.

A=ACTIVE DONOR Anyone who made their first donation more than 12 months ago and has made a donation in the last 12 months.

L=LAPSING DONOR A previous donor who made their last donation between 13-24 months ago.

I=INACTIVE DONOR A previous donor who has not made a donation in the last 24 months. It is people who made a donation 25+ months ago.

S=STAR DONOR STAR Donors are individuals who have given to 3 consecutive card mailings.

Second Byte of code is concerned with FREQUENCY based on the period of recency. The period of recency for all groups except L and I is the last 12 months. For L it is 13-24 months ago, and for I it is 25-36 months ago. There are four valid frequency codes.

1=One gift in the period of recency
2=Two gift in the period of recency
3=Three gifts in the period of recency
4=Four or more gifts in the period of recency

Third byte of the code is the Amount of the last gift.

A=\$0.01 - \$1.99
B=\$2.00 - \$2.99
C=\$3.00 - \$4.99
D=\$5.00 - \$9.99
E=\$10.00 - \$14.99
F=\$15.00 - \$24.99
G=\$25.00 and above

ADATE_2	Date the 97NK promotion was mailed
ADATE_3	Date the 96NK promotion was mailed
ADATE_4	Date the 96TK promotion was mailed
ADATE_5	Date the 96SK promotion was mailed
ADATE_6	Date the 96LL promotion was mailed
ADATE_7	Date the 96G1 promotion was mailed
ADATE_8	Date the 96GK promotion was mailed
ADATE_9	Date the 96CC promotion was mailed
ADATE_10	Date the 96WL promotion was mailed
ADATE_11	Date the 96X1 promotion was mailed
ADATE_12	Date the 96XX promotion was mailed
ADATE_13	Date the 95FS promotion was mailed
ADATE_14	Date the 95NK promotion was mailed
ADATE_15	Date the 95TK promotion was mailed

B. KDD Cup Documents

ADATE_16	Date the 95LL promotion was mailed
ADATE_17	Date the 95G1 promotion was mailed
ADATE_18	Date the 95GK promotion was mailed
ADATE_19	Date the 95CC promotion was mailed
ADATE_20	Date the 95WL promotion was mailed
ADATE_21	Date the 95X1 promotion was mailed
ADATE_22	Date the 95XK promotion was mailed
ADATE_23	Date the 94FS promotion was mailed
ADATE_24	Date the 94NK promotion was mailed
RFA_2	Donor's RFA status as of 97NK promotion date
RFA_3	Donor's RFA status as of 96NK promotion date
RFA_4	Donor's RFA status as of 96TK promotion date
RFA_5	Donor's RFA status as of 96SK promotion date
RFA_6	Donor's RFA status as of 96LL promotion date
RFA_7	Donor's RFA status as of 96G1 promotion date
RFA_8	Donor's RFA status as of 96GK promotion date
RFA_9	Donor's RFA status as of 96CC promotion date
RFA_10	Donor's RFA status as of 96WL promotion date
RFA_11	Donor's RFA status as of 96X1 promotion date
RFA_12	Donor's RFA status as of 96XX promotion date
RFA_13	Donor's RFA status as of 95FS promotion date
RFA_14	Donor's RFA status as of 95NK promotion date
RFA_15	Donor's RFA status as of 95TK promotion date
RFA_16	Donor's RFA status as of 95LL promotion date
RFA_17	Donor's RFA status as of 95G1 promotion date
RFA_18	Donor's RFA status as of 95GK promotion date
RFA_19	Donor's RFA status as of 95CC promotion date
RFA_20	Donor's RFA status as of 95WL promotion date
RFA_21	Donor's RFA status as of 95X1 promotion date
RFA_22	Donor's RFA status as of 95XK promotion date
RFA_23	Donor's RFA status as of 94FS promotion date
RFA_24	Donor's RFA status as of 94NK promotion date
<hr/>	
The following fields are summary variables from the promotion history file.	
CARDPROM	Lifetime number of card promotions received to date. Card promotions are promotion type FS, GK, TK, SK, NK, XK, UF, UU.
MAXADATE	Date of the most recent promotion received (in YYMM, Year/Month format)
NUMPROM	Lifetime number of promotions received to date
CARDPDM12	Number of card promotions received in the last 12 months (in terms of calendar months translates into 9603-9702)
NUMPRM12	Number of promotions received in the last 12 months (in terms of calendar months translates into 9603-9702)
<hr/>	
The following fields are from the giving history file.	
RDATE_3	Date the gift was received for 96NK
RDATE_4	Date the gift was received for 96TK

B. KDD Cup Documents

RDATE_5	Date the gift was received for 96SK
RDATE_6	Date the gift was received for 96LL
RDATE_7	Date the gift was received for 96G1
RDATE_8	Date the gift was received for 96GK
RDATE_9	Date the gift was received for 96CC
RDATE_10	Date the gift was received for 96WL
RDATE_11	Date the gift was received for 96X1
RDATE_12	Date the gift was received for 96XK
RDATE_13	Date the gift was received for 95FS
RDATE_14	Date the gift was received for 95NK
RDATE_15	Date the gift was received for 95TK
RDATE_16	Date the gift was received for 95LL
RDATE_17	Date the gift was received for 95G1
RDATE_18	Date the gift was received for 95GK
RDATE_19	Date the gift was received for 95CC
RDATE_20	Date the gift was received for 95WL
RDATE_21	Date the gift was received for 95X1
RDATE_22	Date the gift was received for 95XK
RDATE_23	Date the gift was received for 94FS
RDATE_24	Date the gift was received for 94NK

RAMNT_3	Dollar amount of the gift for 96NK
RAMNT_4	Dollar amount of the gift for 96TK
RAMNT_5	Dollar amount of the gift for 96SK
RAMNT_6	Dollar amount of the gift for 96LL
RAMNT_7	Dollar amount of the gift for 96G1
RAMNT_8	Dollar amount of the gift for 96GK
RAMNT_9	Dollar amount of the gift for 96CC
RAMNT_10	Dollar amount of the gift for 96WL
RAMNT_11	Dollar amount of the gift for 96X1
RAMNT_12	Dollar amount of the gift for 96XK
RAMNT_13	Dollar amount of the gift for 95FS
RAMNT_14	Dollar amount of the gift for 95NK
RAMNT_15	Dollar amount of the gift for 95TK
RAMNT_16	Dollar amount of the gift for 95LL
RAMNT_17	Dollar amount of the gift for 95G1
RAMNT_18	Dollar amount of the gift for 95GK
RAMNT_19	Dollar amount of the gift for 95CC
RAMNT_20	Dollar amount of the gift for 95WL
RAMNT_21	Dollar amount of the gift for 95X1
RAMNT_22	Dollar amount of the gift for 95XK
RAMNT_23	Dollar amount of the gift for 94FS
RAMNT_24	Dollar amount of the gift for 94NK

The following fields are summary variables from
the giving history file.

RAMNTALL	Dollar amount of lifetime gifts to date
NGIFTALL	Number of lifetime gifts to date
CARDGIFT	Number of lifetime gifts to card promotions to date
MINRAMNT	Dollar amount of smallest gift to date
MINRDATE	Date associated with the smallest gift to date
MAXRAMNT	Dollar amount of largest gift to date
MAXRDATE	Date associated with the largest gift to date
LASTGIFT	Dollar amount of most recent gift
LASTDATE	Date associated with the most recent gift

B. KDD Cup Documents

FISTDATE	Date of first gift
NEXTDATE	Date of second gift
TIMELAG	Number of months between first and second gift
AVGGIFT	Average dollar amount of gifts to date

CONTROLN	Control number (unique record identifier)
TARGET_B	Target Variable: Binary Indicator for Response to 97NK Mailing
TARGET_D	Target Variable: Donation Amount (in \$) associated with the Response to 97NK Mailing
HPHONE_D	Indicator for presence of a published home phone number

(See the section on RFA for the meaning of the codes)	
RFA_2R	Recency code for RFA_2
RFA_2F	Frequency code for RFA_2
RFA_2A	Donation Amount code for RFA_2
MDMAUD_R	Recency code for MDMAUD
MDMAUD_F	Frequency code for MDMAUD
MDMAUD_A	Donation Amount code for MDMAUD

CLUSTER2	Classic Cluster Code (a nominal symbolic field)
GEOCODE2	County Size Code

=====
EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL

INFORMATION LISTED BELOW IS AVAILABLE UNDER THE TERMS OF THE
CONFIDENTIALITY AGREEMENT

EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL EPSILON CONFIDENTIAL
=====