Humboldt University to Berlin

Comparing Approaches for Uplift Modeling of Customer Conversion in Online Stores

Master of Science (M.Sc.)
Information Systems

Chair of Information Systems
Humboldt University to Berlin

By: **Radmer, Tillmann**
Student-ID: 577300

Supervisors: 1. Herrn Prof. Dr. Lessmann

2. Herrn Prof. Dr. Fabian

Submitted on: 04.04.17

Hiermit erkläre ich, dass die Arbeit noch nicht für andere Prüfungen eingereicht wurde, dass sie selbständig verfasst wurde, dass sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, als solche kenntlich gemacht sind und dass bekannt ist, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschung eingeleitet wird.

(Tillmann Radmer)

Table of Contents

1 INTRODUCTION 1

1.1 GOALS 2

2 THEORY OF UPLIFT MODELLING 4

2.1 A/B TESTING 4

2.2 CLASSIFICATION OF MODELING APPROACHES 5

2.3 INDIRECT UPLIFT MODELING 6

2.4 INTEGRATED UPLIFT MODELING 15

3 ASSESSING UPLIFT MODEL PERFORMANCE 17

3.1 CONVENTIONAL METHODS OF MACHINE LEARNING 17

3.2 CUMULATIVE GAINS CHARTS / GINI 20

3.3 QINI CURVES AND COEFFICIENTS 21

3.4 EVALUATION STRATEGY 25

4 EXPERIMENT 27

4.1 DATASET DESCRIPTION 27

4.2 PREPROCESSING 28

4.2.1 Missing Value Imputation 28

4.2.2 Variable Selection 30

4.3 SELECTED MODELS 32

4.4 EVALUATION STRATEGY 36

5 RESULTS 40

5.1 FULL DATASET DISCUSSION 40

5.2 FIRST REDUCED DATASET DISCUSSION 41

5.3 SECOND REDUCED DATASET DISCUSSION 43

5.4 DETAILED ANALYSIS OF APPROACHES 45

5.5 OVERALL ANALYSIS 50

6 CONCLUSION 52

7 BIBLIOGRAPHY I

APPENDIX A: UPLIFT SCORE TABLES V

APPENDIX B: QINI CURVES PLOTS VIII

1 Introduction

The problem of modeling a heterogeneous effect of an action on a subject's behavior arises in many fields. For example, in a direct marketing situation an operator wants to know which customers should be contacted. This is not trivial, as a lot of customers have already decided to buy or cannot be convinced to do so. The increase in customer conversions through the action is called the uplift in the field of direct marketing and models that explicitly model the uplift effect are called uplift models. With the advent of more available data and more powerful machine learning algorithms the field saw an increase in research activity over the past 20 years.

Traditional statistical analysis can determine whether an action has an effect or not. An experiment is conducted in which one group of customers is treated with the direct marketing action – the treatment group – and the other is left unaltered – the control group. Then a statistical test determines whether the conversion rate in the treatment and control group are significantly different – or rather, whether they're not the same. While this is a very useful information for making strategical decisions on whether to continue a marketing campaign, uplift modeling provides a much more sophisticated tool that can even generate an uplift with a treatment for which no overall uplift can be measured with random targeting.

This is possible because of positive and negative effects of the action. Some customers can be convinced by the marketing action and convert when they would otherwise have not, but others might be annoyed and leave the company entirely. (Jaskowski & Jaroszewicz, 2012) With badly designed marketing action those effects could cancel out, leading to the zero-overall uplift. But by identifying the customers who are *not* annoyed by the campaign and instead those that were undecided and change their behavior, an uplift model can make this treatment work successfully. The benefits are especially high if the cost associated with the action is high or if there are significant negative effects of the action for some subjects. In the medical field, this is often the case as many treatments are expensive and can have various negative side effects. In direct marketing the situation is a little different as the costs are usually relatively small –

especially with online marketing campaigns – making targeting a large percentage of the subjects a viable option. However, often there are negative effects through which subjects are "turned off" by the action. Additionally, when the action includes a limited discount, targeting subjects who don't change their behavior incurs costs through lost revenue.

1.1 Goals

The goal of this work is to conduct a comprehensive comparison of approaches for modeling conversion uplift based on which the understanding of their characteristics is advanced. The focus will be on modeling approaches that can be implemented with existing machine learning algorithms. This makes the approaches more accessible for a wide range of applications. In total eight approaches are evaluated and compared. A comparison of all eight approaches in one work has not been done in the literature. Most work focuses on one or two methods, usually the ones introduced in the paper, and compare those to one or two other approaches. Another drawback of other studies is that only one classification algorithm is used for the modeling. Usually this is a logistic regression algorithm, but many more algorithms exist that have been shown to perform better for many tasks. In this work tree-based methods, support vector machines and a naïve Bayes classifier are included in addition to a logistic regression classifier.

For the analysis, the current methods for evaluating uplift model performance are reviewed. The strength and weaknesses are identified and their implementation and interpretation is discussed. Because the current methods don't offer much in the way of explaining an uplift model's performance, a score distribution analysis is introduced. Because many of the approaches discussed work with multiple models internally, those are inspected as well using conventional evaluation methods for machine learning algorithms.

The structure of the work is as follows: First, a thorough introduction to the theory of uplift modelling is provided and the fundamental ideas underlying the different approaches are explained. Strength and weaknesses are identified and some modifications to improve on them are introduced. Second, the different models will be compared in an

experiment based on a real-world dataset. Because of the sensitivity of uplift models to overfitting, variable selection is performed and compared to the full dataset. Third, model performance will be analyzed based on conventional and uplift-specific evaluation methods with the goal of improving the understanding of uplift model behavior.

2 Theory of Uplift Modelling

In uplift modeling the dataset must be obtained by randomly splitting the customers into two groups. On group gets the treatment of which the effect should be modeled. The other group is left unaltered. This is called a randomized controlled experiment and is discussed in the first section of this chapter. The uplift of the dataset is given by

$$U = P(Y = 1|G = T) - P(Y = 1|G = C) \tag{1}$$

$Y = \{0,1\}$ indicates whether a customer converted in the experiment. $G = \{T, C\}$ indicates whether a customer was given the treatment (T) or not (C). This naming convention will be used for those variables throughout this chapter.

After obtaining the data, the goal is to develop a model that learns to rank the customers according to their probability of contributing positively to the uplift.

2.1 A/B Testing

	Respond	No Respond
Treatment	TR	TNR
Control	CR	CNR

Figure 1: The four groups resulting from A/B testing for uplift modeling. (Shaar, et al., 2016)

A/B testing is a term used for a randomized controlled experiment with two variants, the A and the B group. A/B testing is conducted to test if some treatment results in a change in the behavior of the subjects.

In a randomized controlled experiment subjects are chosen at random to be included in either the A or the B group, one of which is not treated. Randomization allows for the greatest reliability and validity of statistical estimates of treatment effects, by reducing the bias. It is equalizing other differentiating factors of subjects across the A and B class that have not been explicitly accounted for in the experimental design by distributing customers with all traits into both groups equally.

The uplift of the experimental data can simply be calculated according to equation (1). Hypothesis testing can be used to reject the H0 hypothesis that there is no difference versus the H1 hypothesis that there is a difference in conversion between the A and the B group. Null-hypothesis testing says something about whether the two distributions are likely the result of sampling from the same distribution. Often one is more interested in the amount of the difference.

Bayesian A/B testing and monte-carlo sampling from the posterior distribution can give estimates of the strength of the uplift effect. The approach starts by modeling the conversion rate in each group. By the central limit theorem, the normal distribution could be used for large sample sizes. Preferable for the for the setting with k successes out of n failures is the Beta distribution with two degrees of freedom (Stich, 2014). The posterior distributions of the two population's conversion probability can be defined as

$$P(Y = 1|G = T) \sim Beta(k_T - 1, n_T - k_T + 1)$$

$$P(Y = 1|G = C) \sim Beta(k_C - 1, n_C - k_C + 1)$$

when assuming a non-informative prior. where k_T and k_C are the number of converters in the treatment and control group respectively. n_T and n_C are the total number of samples in each group. To obtain the posterior distribution of the *difference* in conversion rates samples are drawn from the prior distributions. For each sample pair the difference between the conversion rates is calculated. Then the distribution of the differences approximates the posterior distribution of the uplift between the two groups. (Kurt, 2015)

A hypothesis test for when one is interested in the difference in uplift between two controlled experiments is the χ^2_{net} statistic developed by (Michel, et al., 2014).

2.2 Classification of Modeling Approaches

Uplift modeling approaches can be broadly classified into two groups. Indirect methods on the one hand use one or more standard classification algorithms to predict the conversion probabilities in treatment and control group. Direct methods on the other hand must

develop a classification algorithm that directly optimizes the uplift. (Shaar, et al., 2016)

This work focuses on the indirect methods, but the direct approaches will be briefly introduced at the end of this chapter, because ideas from their conception can help with the understanding and interpretation of uplift modeling results.

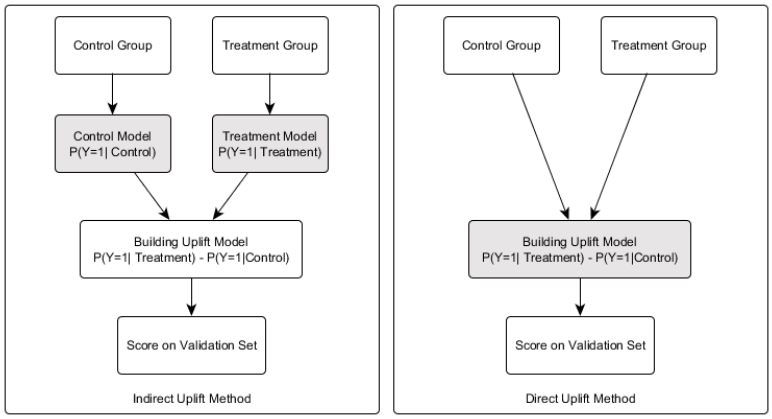

Figure 2: Direct Vs. Indirect Uplift Modeling (Shaar, et al., 2016)

2.3 Indirect Uplift Modeling

Most of the indirect methods are different approaches to obtain an estimate for the class conditional conversion probabilities given the information x on the customers. The uplift probability is then the difference between those two probabilities:

$$U(x) = P(Y = 1|\mathrm{G} = T, x) - P(Y = 1|\mathrm{G} = C, x) \tag{2}$$

Response Modeling

A response model is any binary classification algorithm that is trained to predict whether a customer will respond or not. This can be written as

$$U(x) = P(Y = 1|x) \tag{3}$$

The typical approach to using response models to model heterogeneous treatment effects is to predict the response in the group that was treated.

$$U(x) = P(Y = 1|G = T, x) \tag{4}$$

The control group from the randomized controlled experiment is ignored completely in this process. (Radcliffe, 2007) Response models are widely

used and studied in the academic literature. There are applications using artificial neural networks (Tang, 2011; Olson & Chae, 2012), support vector machines (Shin & Cho, 2006; Kim, et al., 2008),and decision trees (Olson & Chae, 2012).

In general, finding the group that is most likely to respond seems like a good idea. But this ultimately fails to capture the true effect, because it ignores that some customers would respond with or without the treatment.

An interesting aspect that has been studied in the response modeling literature is that of the class imbalance problem. The problem is that of one class being much more frequent in the data than the other. This is the case for response modeling and uplift modeling were there are usually many more non-responders than responders. The consequence is that the classification algorithm optimizes more for the majority class, which can result in poor performance. Under-sampling of the majority class and over-sampling of the minority class are two ways to overcome the use. (Crone, et al., 2006) found that under-sampling significantly decreases out-of-sample performance. The risk is that because under-sampling disregards a lot of samples it might be biased and can distort the distribution of non-responders. To overcome this issue (Kang, et al., 2012) have proposed an algorithm that first clusters the non-responders and samples from the clusters in proportion to the cluster-size. This process can be repeated multiple times to generate several training datasets on which an ensemble of response models can bet trained.

While only feature selection and missing value imputation are addressed in this work, it is important to keep in mind that the model performance can still improve with further pre-processing of the data. Dealing with the class imbalance could not only benefit the response model, as the Naïve Two Model Approach, the Naive One Model Approach, the Interaction Term Approach, and the generalized Class Variable Transformation Approach depend on the classifying converters against non-converters, too.

Naïve Two Model Approach

A simple path to modelling the uplift is to use two response models. One models the response rate in the treatment group and one in the control

group. Hence, each classifier produces a group conditional response probability which can be used in equation (2). The predicted uplift score is the difference between the two classifier predictions.

$$U(x) = P_1(Y = 1|G = T, x) - P_2(Y = 1|G = C, x) \tag{5}$$

(Hansotia & Rukstales, 2002) note that "the resulting model scores should be an accurate forecast of the future behavior, since they are not being used merely to rank-order customers for selection". They compute the lift for deciles of customers on a holdout dataset and fit a polynomial model for smoothing. Then the output scores are mapped to lift scores using this model before computing the final uplift score with equation (5). This approach was not implemented in this work, but model score calibration using Platt-scaling was tested (Platt & others, 1999).

Interaction Term Approach

This approach introduced by (Lo, 2002) models the differential treatment effect by including the group variable $G = \{C, T\} = \{0,1\}$ which indicates whether a sample is from the control or treatment group. Then additional interaction term covariates $x \cdot G$ are introduced, where x is the matrix of normal features. The matrix of interaction term covariates is

$$x^*(G) = x \cdot G \tag{6}$$

and the model is any predictive model f that is given the origianl convariates, the group indicator and the interaction covariates:

$$Y(x) = f(x, G, x^*) \tag{7}$$

This way a single model can learn differences in conversions between the treatment and control group. For readability reasons the dependency of x^* on G is not explicitly written out.

For prediction, each sample is predicted two times: Once with the group variable and interaction term variables set to the treatment group and once set to the control group. The difference between the two predictions is the predicted uplift score:

$$U(x) = f(x, G, x^*)|_{G=1} - f(x, G, x^*)|_{G=0} \tag{8}$$

With $G = 1$ indicating the treatment group and $G = 0$ indicating the control group. In the paper (Lo, 2002) evaluate the approach using simulated data from a logistic model with four correlated features. The treatment and control classes are sampled uneven with 80% from the treatment group and 20% from the control group, which is a realistic assumption for modelling online marketing campaign data. However, (Lai, et al., 2006) note that the generated data has unusually high response rates. The dataset used in this work is not synthetically generated and has a much lower response rate. (Lai, et al., 2006) further point out that the model assumes that the indicator T is correlated with the response variable.

> *"This implies that the campaign from which such data was collected has successfully targeted undecided customers. In other words, a successful model has been deployed in the previous campaign, which defeats the purpose of constructing another such model from the training data." – (Lai, et al., 2006)*

While it does require that some undecided customers changed their behavior, it does not require a model that successfully identifies such. Even in a completely randomized controlled experiment some undecided customers will be in the treatment group. If the treatment is effective – that is, when given to an undecided customer it has a positive impact on the conversion probability – a positive treatment effect will be present and statistically significant in the data, given a large enough sample size. If this is the case the conversion variable will be correlated with the group variable, even though it is probably less so than when a successful model was used for the experiment.

In their work (Tian, et al., 2014) propose a similar approach for estimating a medical treatment effect. They consider the response variable $Y = \{0,1\}$ indicating a negative and positive response respectively. The group indicator variable is $G = \{-1,1\}$ with 1 indicating a treatment. This allows to fit a logistic regression just on the interaction covariates instead of augmenting the covariates, because in the latter case the covariates are all 0 for $G = C$. The interaction covariates are defined as

$$x^* = x \cdot \frac{G}{2} \tag{9}$$

They show that a logistic regression model

$$P(Y = 1|x^*) = \frac{\exp(\beta \cdot x^*)}{1 + \exp(\beta \cdot x^*)} = \frac{1}{1 + \exp(-\beta \cdot x^*)} \tag{10}$$

for the conversion variable models the differential treatment effect even when the model is not correctly specified.

$$P(Y = 1|x^*, T) - P(Y = 1|x^*, C) = \frac{\exp(\beta \cdot x^*) - 1}{\exp(\beta \cdot x^*) + 1} \tag{11}$$

In this work, the approach as presented by (Lo, 2002) is investigated and referred to as the interaction term approach or ITA.

Class Variable Transformation Approach

The class variable transformation approach also enables uplift modelling with only a single classification model and was developed by (Lai, 2006; Lai, et al., 2006). The approach is based on the idea that there are groups of customers with a positive or negative impact on uplift. Each of the groups from Figure 1 is assigned a binary label, based on whether it potentially contributes to an increased conversion when treated.

Lai identifies three groups of customers that can be found in one or more fields from the A/B experiment:

1. Decided
2. Undecided
3. Non-converting

The decided customers will buy regardless of the stimuli and targeting this group does not generate any additional conversions. The undecided customers have an increased purchasing probability if they get a promotion and not if they don't receive one. Hence, targeting this group will generate additional conversions. The last group will not buy regardless of the stimulus. Like the decided group, targeting this group will not generate additional conversions. (Lai, et al., 2006, p. 2)

	Respond	No Respond
Treatment	TR	TNR
Control	CR	CNR

Figure 3: "Lai's approach contingency table of uplift modeling; gray areas represent positive uplift; white areas represent negative uplift." – (Shaar, et al., 2016)

Figure 3 the positive (gray) and negative classes (white). The TR group contains both decided and undecided customers. Only the undecided ones are convinced by the campaign and contribute to the additional conversions. The CNR group also contains some undecided customers, but instead of decided customers contains non-converting customers who will never convert. These two groups are the only ones containing undecided customers, so they form the positive class (PC). The CR group exclusively contains decided customers, while the TNR group only contains non-converting customers. In both cases targeting cannot improve the conversion rate. These two groups form the negative class (NC). The transformed target variable is defined as

$$Y^* = \begin{cases} 1 \; if \; G = T \; and \; Y = 1 \\ 1 \; if \; T = C \; and \; Y = 0 \\ \quad 0 \; otherwise \end{cases} \tag{12}$$

Then a classification algorithm is trained to predict the probability of the positive class. During prediction, the uplift score is computed as the difference of the predicted probability for the positive and negative classes:

$$U(x) = P(TR + CNR|x) - P(TNR + CR|x) \tag{13}$$

Hence, if PC and NC are predicted to be equally likely the uplift score will be zero.

(Jaskowski & Jaroszewicz, 2012) and (Weisberg & Pontes, 2015) have shown independently that the model trained on the transformed target variable from equation (12) models the uplift probability as

$$U(x) = 2 \cdot P(Y^*|x) - 1 \tag{14}$$

One upside of the class variable transformation approach is, that besides only requiring one classification model, it effectively eliminates the class

imbalance problem that arises in many direct marketing settings. Since the general response effect is small relative to the whole population, the TR and CR groups are small compared to the TNR and CNR groups. As described before this can lead to poor model performance by making the model overly confident in predicting the majority class. By combining the small TR group with the large CNR group, and likewise the CR with the TNR group, this problem is eliminated. (Lai, et al., 2006, p. 4)

Additionally, the authors suggested that one can boost the importance of decided customers in the negative class (CR+TNR) by oversampling CR, giving their "characteristic" more weight in the NC compared to the PC. Thus, the decided customers are given a greater probability of belonging to the NC class and will more cancel the TR class in the PC which also contains a lot of decided customers. Conversely, one can under-sample the CNR, limiting the number of non-converters in the PC. This can be beneficial when focusing on conversions as the CNR group contains non-converting customers. (Lai, et al., 2006, p. 4)

The class variable transformation approach has been extended by (Pechyony, et al., 2013) to jointly optimize the incremental gain and the total number of conversions. The motivation for this extension is that for advertisers maximizing the incremental gain is the optimal strategy. But for other users, for example publishers who are paid by the click, maximizing the total conversions might give higher revenue. On a dataset with two advertisers their approach results in a total number of conversions that is comparable to a pure purchase maximizing model, while also giving high incremental gains. They compare their model to the "difference of purchase models" approach by (Lo, 2002).

The generalised formulation was derived by (Kane, et al., 2014):

$$U(x) = \frac{P(T,R|x) - P(T,N|x)}{P(T)} + \frac{P(C,R|x) - P(C,N|x)}{P(C)} \quad (15)$$

Instead of a binary classifier, a multi-class model is trained on an indicator variable for each of the four fields from Figure 1. Instead of this generalized model, one can weigh the positive and negative score in equation (13) by their class probability as done by (Shaar, et al., 2016):

$$U_W(x) = P(Y^* = 1|x) \cdot P(Y^* = 1)$$

$$-P(Y^* = 0|x) \cdot P(Y^* = 0) \tag{16}$$

Both the generalized approach and the weighted approach are tested in this work. The former is referred to as 'Kane' and the latter is called simply the class variable transformation approach (CVT).

Naive One Model Approach

To arrive at equation (15) (Kane, et al., 2014) rewrote equation (2) in two different ways, by replacing either of the two conditional probabilities by their inverse and added the two versions together. Instead of this step one can rewrite the conditional probabilities in terms of the marginal probabilities and the class prior probabilities.

$$Uplift = P(R|T,x) - P(R|C,x)$$

$$= \frac{P(R,T|x)}{P(T|x)} - \frac{P(R,C|x)}{P(C|x)}$$

$$= \frac{P(R,T|x)}{P(T)} - \frac{P(R,C|x)}{P(C)} \tag{17}$$

In the last step, the assumption is made that the class probabilities are independent of x. This holds if the the assignment to treatment and control groups is completely randomized. The difference of this formulation compared to equation (15) is that the undecided customers in the CN group and the non-response customers in the TN group are not modelled explicitly as positive and negative respectively.

Reflective & Pessimistic Uplift Modeling

In their work (Shaar, et al., 2016) identify overfitting, noise modeling, variable correlation and disturbance as a major problem when doing uplift modeling. They propose an approach that is more resilient to those effects while still achieving good prediction scores. A weighted version of (Lai, et al., 2006) class variable transformation approach is used alongside a new approach called *Reflective Uplift*. The authors identify two disturbance effects that could reduce the performance:

1. **Response effect**
   This effect exists because responders are usually similar and some variable are correlated with the response variable.
2. **Partitioning effect**
   This effect exists when the group assignment is not independent of the features, e.g. the group variable is correlated with the features.

With proper sampling in the controlled experiment the second effect should be minimal and employing bagging algorithms should further help reduce the impact. The first effect however cannot be reduced with such measures.

When one or more of the variables is highly correlated with the conversion variable. Converters are usually similar so this correlation would then be similar across treatment and control group. Instead of modeling the differential treatment effect a model will consequently model the conversion effect in each group which will lead to near-zero uplift predictions, even when there is a positive effect. This is one reason the weighted class variable transformation approach by Lai was chosen. By mixing converters with non-converters into a positive and negative class the approach becomes more resilient against the response disturbance effect.

$$U_{Lai}(x) = P(Y^* = 1|x) \cdot P(Y^* = 1) - P(Y^* = 0|x) \cdot P(Y^* = 0) \tag{18}$$

The class variable transformation is the same as in equation (11).

The *Reflective Uplift* approach specifically tries to avoid the response effect at the cost of being more susceptible to the partitioning effect. This is justified with the fact that the latter effect can be controlled for in the A/B experiment. Two models are build, but unlike the naïve two model approach, they don't learn the conversion effect in each group. Instead the two models learn the "treatment effect" in the converter and non-converter group separately. Then the uplift is calculated, whereas the groups are treated as positive and negative like in the class variable transformation approach:

$$U_{pos}(x) = P_{M1}(G = T|Y = 1, x) \cdot P(G = T, Y = 1)$$

$$+P_{M2}(G = C|Y = 0, x) \cdot P(G = C, Y = 0)$$

$$U_{neg}(x) = P_{M2}(G = T|Y = 0, x) \cdot P(G = T, Y = 0)$$

$$+P_{M1}(G = C|Y = 1, x) \cdot P(G = C, Y = 1)$$

$$U_{Reflective}(x) = U_{pos}(x) - U_{neg}(x) \tag{19}$$

P_{M1} and P_{M2} are probabilities obtained from the two models respectively. The final uplift score is the average of the weighted version of Lai's method and the *Reflective Uplift* approach:

$$U_{Pessimistic}(x) = \frac{1}{2}\left(U_{Lai}(x) + U_{Reflective}(x)\right) \tag{20}$$

Both the full pessimistic uplift approach and the reflective uplift approach are tested in this work and referred to as pessimistic and reflective, respectively.

2.4 Integrated Uplift Modeling

Instead of fitting two models to predict the response rate in each group separately and subtracting the results or transformation of the target variable, the direct methods fit a model directly on the uplift of subgroups of the population. Integrated uplift modeling approaches have been developed based on support vector machines (SVM) (Zaniewicz & Jaroszewicz, 2013; Jaroszewicz & Zaniewicz, 2016) and tree-based algorithms. The latter are introduced shortly in this chapter and their relation to the indirect approaches above is explored.

Tree-Based Approaches

When training an integrated approach for uplift modeling one problem is that the uplift is not defined at the level of an individual. Many algorithms need the knowledge on what the exact, optimal outcome for any individual is. Tree based methods are an exception as they're inherently based on groups and sub-groups of samples. When finding a split, the tradeoff is usually between maximizing the difference in the outcome and minimizing the difference in size between the two halves of the split. (Radcliffe & Surry, 2011)

For growing an uplift tree (Hansotia & Rukstales, 2002) directly use the uplift in the subpopulations, but ignore the population size tradeoff. (Radcliffe & Surry, 2011) introduce a significance based splitting criterion which trains a linear model for every potential splitting point. The linear model learns to predict the conversion rate based on the group the sample is in, whether it is left or right of the split, and an interaction term that captures the relationship between the two. This last term is the one that is used to define the goodness of a splitting point. The final metric is derived from the t-statistic of the interaction term.

(Rzepakowski & Jaroszewicz, 2010) developed a splitting criterion based on the Kullback-Leibler (KL) divergence and squared Euclidian distance. The distribution divergence is used as a measure of how much two probability distributions differ. The difference between the conversion rate in the treatment and control groups of the split is modeled through either of the two divergence criteria mentioned above. (Rzepakowski & Jaroszewicz, 2012) They show that the squared Euclidian distance performs superior to the KL divergence. Finally, the divergence is normalized such that splits that would results in very unequal treatment to control ratios are penalized. This approach can be extended to random forests, e.g. growing multiple bagged trees and averaging their results. (Guelman, et al., 2015) The resulting trees are however susceptible to overfitting and have a bias to select features with many possible splits when dealing with categorical data. The causal conditional inference forests developed by (Guelman, 2015) overcome those issues. The uplift random forests are tested empirically with good results and it is shown that increasing the size of the random forest increases model performance. (Sołtys, et al., 2015)

Uplift trees map an input from a set of properties to an empirical uplift score obtained from the subpopulation of training samples with similar properties. This is not so different from any of the approaches indirect approaches from above when the smoothing post-processing step by (Hansotia & Rukstales, 2002) is applied. The models learn to map a customer based on some criteria into a one-dimensional uplift space. The smoothing step clusters the approaches into subgroups for which the empirical uplift is calculated similar to the subgroups found in the uplift tree algorithms.

3 Assessing Uplift Model Performance

Assessing how well a model can predict the heterogeneous treatment effect is not an easy task. The reason for this is, that the it is not possible to measure the behavior of individuals in both the treatment and control group. One can only ever measure an individual in either one of the two groups. Consequently, it is not possible to calculate an error for every sample in the dataset and compute some statistics over it to come up with an estimate of model performance. Instead one must consider the samples as a group and find a metric that relates the model predictions to the uplift performance. The metrics that are obtained in that way and their characteristics are explained in the second half of this chapter. In the first half, more conventional techniques for assessing model performance will be discussed with regards to uplift modeling.

3.1 Conventional Methods of Machine Learning

Generally, when evaluating a binary classifier there will be four types of predictions:

- True positives (TP):
  The model correctly predicts the positive class.
- True negatives (TN):
  The model correctly predicts the negative class
- False positives (FP):
  The model classifies a negative sample as positive.
- False negatives (FN):
  The model classifies a positive sample as negative.

With those the true positive rate (TPR), true negative rate (TNR), false positive rate (FPR) and false negative rate (FNR) can be defined (see Table x). The TPR is also called the sensitivity or recall and measures how many of the positive samples are classified as such. The TNR is also called the specificity or precision and measures the proportion of negative samples that are correctly identified. The FPR rate is also referred to as the “Type 1” error, while the FNR rate is called the “Type 2” error (see Table 1).

The accuracy is defined as the percentage of correct predictions on the total number of predictions made. This is a good measure to quickly summarize classifier performance in one metric, but especially when dealing with imbalanced datasets this can be misleading. If one class is much larger than the other one, the model will perform very well by classifying most instances into the majority class at the expense of correctly identifying samples in the minority class.

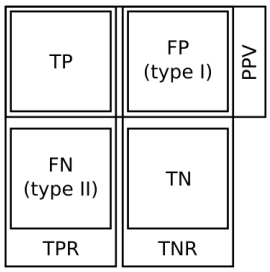

Figure 4: The true and false predictions in the confusion matrix.

A good visualization of the TP, TN, FP and FN metrics is the confusion matrix, which allows to easily detect if the classifier is biased in its error rate. By plotting the four metrics in two times two matrices with the number of correct predictions on one diagonal and the number of false predictions on the other, one can easily see whether the classifier makes significantly more mistakes in one class or the other.

Another metric for model accuracy is the F_1 score. It is the harmonic mean of the PPV and TPR:

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \tag{21}$$

This is a special case of the F_β score which weights the PPV against the TPR:

$$F_1 = (1 + \beta^2) \cdot \frac{PPV \cdot TPR}{(\beta^2 \cdot PPV) + TPR} \tag{22}$$

Specifically, the F_2 and $F_{0.5}$ score weight the TPR and PPV higher respectively.

While all the conventional metrics and methods don't give a direct indicator of the uplift performance of a model, they can be a good starting point to understanding why uplift models might not perform as

expected. When a classifier inside one of the approaches is not able to discriminate the classes in the target variable, this could be a reason for the poor model performance. This is used for "debugging" of models throughout the experiment conducted for this work. Examples of the insights that can be gained this way will be given in the discussion of the results (Chapter 5).

Table 1: Error rates defined across the binary classification results.

Name		**Equation**
True Positive Rate (TPR)	Sensitivity, Recall	$TPR = \frac{TP}{TP + FN}$
True Negative Rate (TNR)	Specificity	$TNR = \frac{TN}{TN + FP}$
False Positive Rate (FPR)	Type I Error	$FPR = \frac{FP}{FP + TN} = 1 - TPR$
False Negative Rate (FNR)	Type II Error	$TPR = \frac{FN}{FN + TP} = 1 - TNR$
Positive Predictive Value (PPV)	Precision	$PPV = \frac{TP}{TP + FP}$
Accuracy (ACC)		$ACC = \frac{TP + TN}{TP + TN + FP + FN}$

It should be noted that the receiver operator characteristic (ROC) was also considered, but due to the size of the dataset it is not trivial to compute. Specifically, the cumulative sum overflows for large datasets, even when the target variable is a binary outcome. In Python's numpy library the cumulative sum of 110,000 ones overflows, while 100,000 ones don't:

Table 2: Example of numerical overflow in the numpy implementation of cumsum.

input size	**result**
100,000	$[1e^0, 2e^0, \dots, 1e^5]$
110,000	$[1,2,1, \dots, 2,1,1]$

3.2 Cumulative Gains Charts / Gini

A widely-used method to assess model performance in all kinds of ranking tasks is the cumulative gains chart, based on which the Gini is defined. It doesn't consider the two groups that exist in A/B experiments. The gains chart is constructed by first sorting the evaluation dataset samples by the model predictive score. Next, the samples are segmented into quantiles and in each quantile the number of positive samples is counted. Finally, plotting the cumulative positive samples gives the cumulative gains chart. An alternative is to calculate the percentage of the total conversions achieved in each quantile. The y-axis is then scaled from 0 to 1 and the scaling of the curve is independent of the population size making comparisons easier.

The ideal model would place all positive samples exclusively in the top quantiles by scoring them higher than negative samples. This would result in a cumulative gains chart that raises from zero to the total number of positive samples in the first quantiles. After reaching the maximum the curve would stay flat. Random targeting would distribute the positive samples equally across all quantiles such that the cumulative gains chart turns into a straight line going from zero positive samples to the maximum number of positive samples at 100% (see Figure 5). (Radcliffe, 2007)

Based on the described curves the Gini score is a metric for how close the tested model is to the ideal model. First, the area between the model curve and the random targeting line, and area between the optimal curve and the random targeting line is calculated. Then the ratio of the two areas gives a score that lies in the interval $[-1,1]$, or between -100% and 100% of the optimal gains curve. The score can be negative if the model places more negative samples in the top quantiles than random placement would, resulting in a gains curve that is under the random targeting line and with a "negative" area. (Radcliffe & Surry, 2011; Radcliffe, 2007) If the percentual gains curve is used for the calculation the Gini score will be independent of the population size.

$$g_0 = \frac{A_{gain} - A_{random}}{A_{optim} - A_{random}} \tag{23}$$

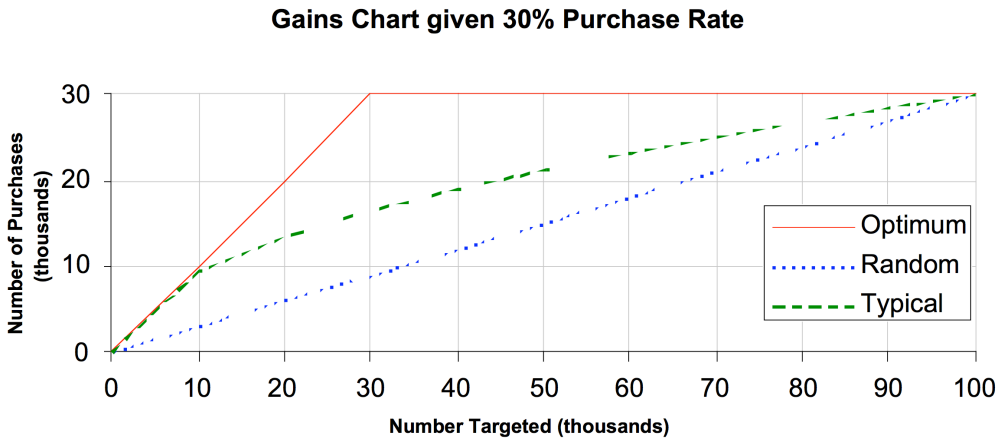

Figure 5.: Cumulative gains chart example on a theoretical dataset with a conversion rate of 30%. The optimal gains chart rises at the maximum rate towards until all conversions are realized, after which it stays flat. (Radcliffe, 2007)

3.3 Qini Curves and Coefficients

The cumulative incremental gains curves, "gains chart for uplift" or qini curves were first introduced by (Radcliffe, 2007) and were used in several papers since then. They're an extension to the cumulative gains chart presented in the previous section that takes both groups from the A/B experiment into consideration. Instead of the total number or percentage of conversions achieved, they consider the difference in conversions between the treatment and control group within each quantile. The uplift score for a quantile is the cumulative uplift up to that point. Hence, the vertical axis doesn't show the number of cumulative realized conversions, but the number of cumulative realized *incremental* conversions.

When calculating the difference in conversion between the two groups in each quantile the question arises which samples should be matched witch each other. The idea is that the model is used to rank both the treatment and control group samples. Then the highest scoring samples from both groups are matched together. Two ways of calculating the curves are possible:

1. The quantiles for the segments are taken from the joint score distribution across both groups.
2. The quantiles are computed for each group individually.

Theoretically, both methods have advantages and disadvantages with regards to the usefulness of the uplift curves:

- **Comparability**
  In the first approach the population sizes could vary from segment to segment. This happens when the groups are dissimilar enough, that the model ranks them unequally across the score range. For the first approach weighting the absolute number of conversions to match the population sizes is mandatory.
- **Interpretability**
  The second approach doesn't suffer from the comparability issue outlined above, but at the expense of interpretability. This also happens when the scores across the treatment and control group are sufficiently different.

  > *"This solution is not ideal as there is no guarantee that the highest scoring examples in the treatment and control groups are similar, but it works well in practice." – (Rzepakowski & Jaroszewicz, 2012, p. 314)*

  Intuitively, the uplift curve should say '*the model will generate an uplift of X when using the cutoff at the Y% quantile*'. To validate the above statement, one would conduct an A/B experiment where a percentage of the customers selected for treatment by the model are *not* treated. Then the uplift between the two groups in the newly conducted experiment should be close to the uplift at the cutoff in the qini curve. With the second method, this cannot be said because the choice of the cutoff is arbitrary. One can choose the one from either the treatment or control group and there is no rational argument that can be made for either one. Then the samples from one of the groups in the experiment come from different score ranges as were used for computing the qini curve.

Of course, the interpretability issue is only significant if the score distribution between the two groups are significantly different. This does not seem to be an issue in practice and comparing the two ways of computing the qini curves during experimentation supports this (Figure 6). It is safe to assume that they're similar, because the uplift effect is

usually small compared to the main conversion effect. As we will see when discussing the results, this is the case with the dataset that is used in this work (Chapter 4.1 & 5).

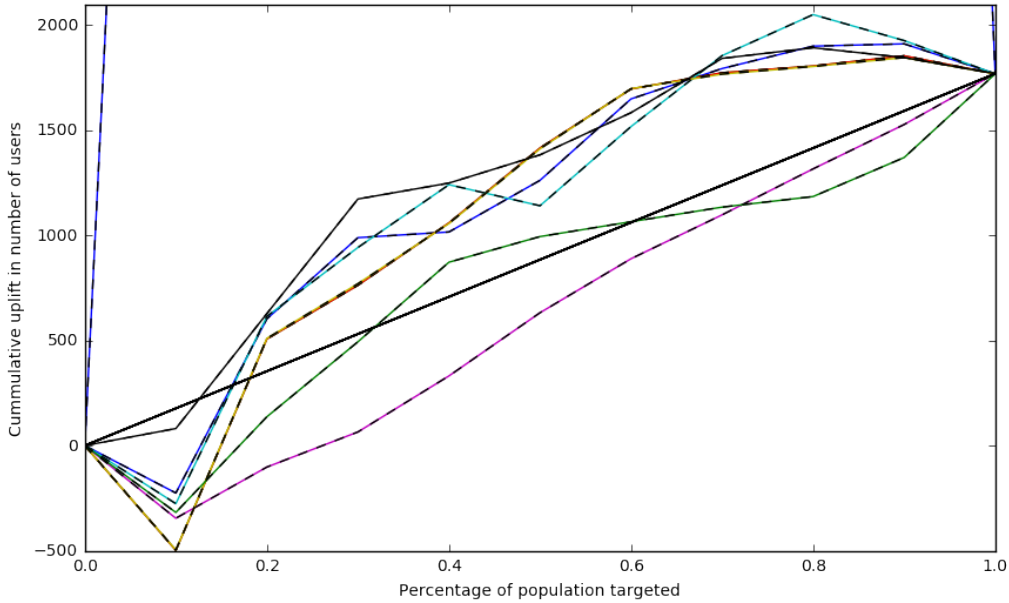

Figure 6: Uplift curves calculated by both methods for all approaches tested in this work. The black dashed curves are computed using the joint global quantile, the colored lines are computed using separate quantiles for the treatment and control group.

Since the size of the treatment and control group can end up differently in each quantile or across the whole dataset, the calculation of the incremental gain must be adjusted for population size (Radcliffe, 2007):

$$U_s = R_t - \frac{N_t}{N_c} \cdot R_c \tag{24}$$

R_t and R_c are the number of positive classes in the segment s. N_t and N_c are the number of treatment and control samples in the same segment.

Unlike in response modeling, the uplift model can exhibit negative effects through customers that would have bought had they not been treated. Because the qini curve is a direct visualization of the cumulative treatment effect, it is possible that the curve raises above the uplift of the dataset before converging to the overall uplift at the 100% quantile (Figure 6). If that is case, the uplift model put more samples whose behavior is positively changed by the treatment in the first quantiles and samples who were adversely affected in the later quantiles. An interesting observation is, that in this way a model can generate uplift across

segments of the qini curve even though the dataset has an *overall uplift of 0%*.

A theoretically optimal qini curve is needed to compute a qini score comparable to the gini score, which is a little bit more intricate than before. In the best scenario, all positive samples in the treatment group are scored higher than any positive sample in the control group. In this case, the uplift will be

$$\begin{aligned} U_s &= R_t - \frac{N_t}{N_c} \cdot R_c \\ &= R_t - \frac{N_t}{N_c} \cdot 0 \\ &= R_t \end{aligned} \tag{25}$$

for all top segments without positive samples in the control group. The maximum positive uplift is thus the conversion rate in the treatment group. The model uplift cannot exceed the dataset uplift by more than the conversion rate in the control group. If all the positive samples in the control group are similarly concentrated in the worst quantiles, the qini curve will take the optimal form shown in Figure 7. When the class response rates are very high, the uplift is limited by the limitation to separate the responders in each class into opposing quantiles. The converters in each group must inevitably overlap in the middle quantiles reducing the uplift at those points. (Radcliffe, 2007)

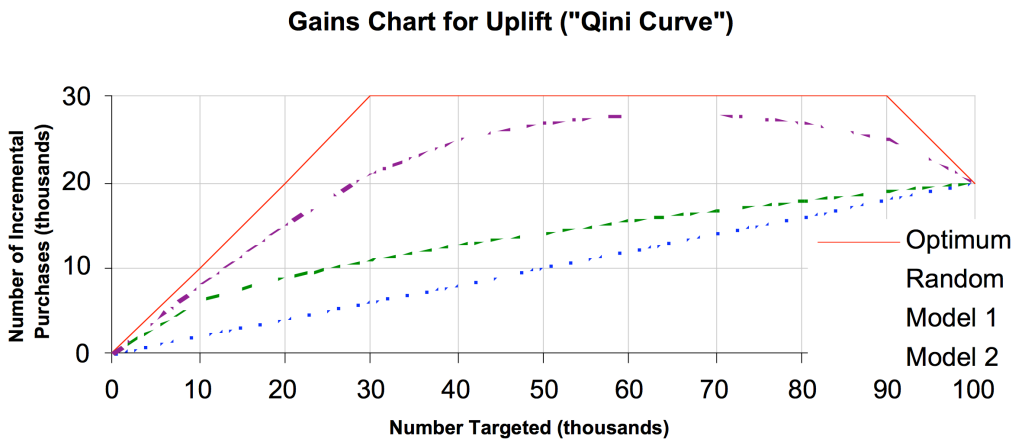

Figure 7: The optimal qini curve for a dataset with an overall uplift of 20%. The conversion rate in the treatment group is 30%, which is the maximum the qini curve can reach. The conversion rate in the control group of 10% brings down the qini curve to the overall uplift. (Radcliffe, 2007)

With that it is possible to compute the analogue to the gini denoted as q_0. However, since the theoretical optimum can be orders of magnitudes larger than the achievable qini curve, this way of scaling the area score is not recommended by (Radcliffe & Surry, 2011).

$$q_0 = \frac{A_{model} - A_{random}}{A_{optimal} - A_{random}} \tag{26}$$

They suggest the unscaled qini measure Q instead, which is simply the area between the qini curve and the random targeting line.

$$Q = A_{model} - A_{random} \tag{27}$$

This can be normalized with the population size N by dividing with N^2, which is useful for comparing between experiments.

$$Q_{norm} = \frac{Q}{\frac{N^2}{2}} = \frac{2 \cdot Q}{N^2} \tag{28}$$

The normalized Qini of the random targeting line for a dataset with 1,000,000 customers with an overall uplift of 10,000 customers (1%) would be

3.4 Evaluation Strategy

Looking at the qini score alone is often not good enough to accurately assess uplift model performance. Depending on the intended use case different criteria can be used additionally. (Kane, et al., 2014) and (Radcliffe & Surry, 2011) propose additional metrics to be considered alongside the overall qini score:

1. ***15% qini score /cutoff performance***
   This is useful for campaigns with a limited budget. Generally, if a cutoff constraint is known, either from a cost analysis from a previous experiment or due to practical reasons, the most relevant metric is the uplift at that cutoff point. Otherwise, targeting all customers up to the cutoff where the qini curve starts to fall is better.

2. R^2 ***regression score / monotonicity of incremental gains***
   (Kane, et al., 2014) use the R^2 as a measure for the models stability. The R^2 is taken from a linear regression onto the segment uplifts. (Radcliffe & Surry, 2011) similarly look for incremental gains that are monotonically falling from the highest gain in the first quantile to the lowest gain in the last quantile. A linear decline is not always the case and even a good model might not have a linear distribution of the segment-wise uplifts as the effect might well be non-linear.
3. **Maximum impact**
   When the monotonicity requirement is met and the data exhibits negative effects, the model with the higher maximum uplift is considered better. (Radcliffe & Surry, 2011)
4. **Tight validation**
   A model is more robust if the patterns in the incremental gain are similar across the training, test, and validation data. (Radcliffe & Surry, 2011) note that the validation patterns are more different than in conventional predictive modelling due to the inherently high errors in uplift modeling.

The evaluation procedure followed in this work is described in detail in Chapter 4.4.

4 Experiment

4.1 Dataset Description

The experiment is based on a large dataset with samples from multiple (27) experiments on online retailer sites. It consists of 3051990 samples with 62 features. The data was collected between October 9. and November 20. 2016. The features are metrics collected about a user's browsing behavior on the site.

The dataset is imbalanced between the two groups as there are about three times as many samples in the treatment groups as in the control group (see Table 3). The main response effect is above 7% in either group. Compared to the uplift between the two groups of about 0.2% this is large, which is a common problem in uplift modelling, as most model will pick up on the main effect instead of focusing on the incremental uplift.

Table 3: Imbalance between treatment and control group

	number		**conversions**	**rate**	**uplift**
Control	766155	(25.1%)	57285	7.47%	0.21%
Treatment	2285835	(74.9%)	175791	7.69%	
Total	3051990				

For the analysis of the uplift effect, the response rate in each group was modeled with a Beta distribution with an uninformative prior distribution – e.g. $\alpha = 0, \beta = 0$. With 1,000,000 samples from each distrubtion the approximate likelihood of the control group having a greater response rate than the treatment group is 0.0%, e.g. none of the samples had a greater control response rate than treatment response rate. This is not surprising given the large sample count in the dataset. Computing the difference between the samples drawn from the modelled treatment and control distributions gives the posterior distribution of the differential treatment effect. The approximate two-sided 95% confidence interval is $[0.15\%, 0.28\%]$ (see Figure 8).

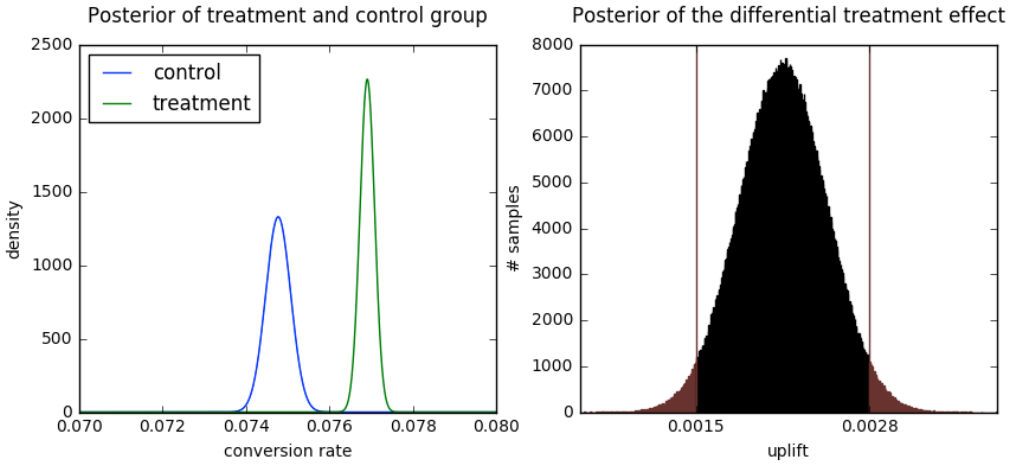

Figure 8: Left: Modeled posterior distributions of the treatment and control group response rate. Right: Approximate sampled posterior distribution of the differential treatment effect between the two groups. The vertical red lines mark the upper and lower bound for the 95% confidence interval.

4.2 Preprocessing

4.2.1 Missing Value Imputation

Many features in the dataset contain missing values which need to be handled before training. The dataset contains 28 variables with missing values many of which have a large percentage of samples without values. Hence, imputation is very desirable not only to because most classification models can't handle missing values (see Table 5).

There are many different strategies for dealing with missing values. Generally, a careful analysis of the variables and their causal impact on the target variable must be conducted before selecting a strategy for handling the missing values. Often, a missing value carries some information. For example, a customer might not have a certain field set in the database because she left that field open (if it is optional). This can say something about the customer and their most likely behavior.

Since there are variables with many missing values, a simple approach would be to drop variables with missing values. However, a lot of potentially useful information is lost, which might reduce model performance. Another approach is univariate imputation using some statistic of the available values for the variable. Examples are the mean or median imputation. This is an easy fix and very fast to compute, but doesn't work well for those variables that have many missing values.

More importantly this method doesn't consider anything that we know about that sample. The sample is treated as the average sample, which at best adds very little information and at worst reduces the information of other samples that have values close to the mean/median.

This can be overcome by using multivariate techniques. A common approach is to build a predictive model for each column with missing values that is trained on the features without missing values. Once all features are imputed at least once, one can use all variables to build the imputation models and iterate over all variables until convergence, e.g. until the imputed variables don't change. This can potentially add information to samples with missing values by learning from all the samples available. Note, that this technique assumes that the features are not independent of each other, such that the correlations can be exploited. This runs against the assumptions of many statistical models. Furthermore, if the variable selection is done carefully, features with redundant information would be eliminated. When using a multivariate imputation in those cases, the learned imputation model is likely modelling the noise in the date and not a real relationship. Another disadvantage is the iterative process of building many models can be very computationally expensive if the number of features and samples is large or if the imputing model has long training times. Since the features are highly correlated no multivariate imputation technique was used.

In any of the mentioned imputation techniques the information about which sample was missing which features is lost. But as stated earlier this information can indeed be very valuable in the modelling stage and can improve model performance. One way to include this information is to create binary variables for each column with missing values that indicate whether the sample was missing the values or not. In our dataset with 28 columns with missing values that would result in 28 extra features. However, an analysis of the missing values suggests a structure in the missing values. Groups of seemingly causally related variables share the same number of missing values and comparing the samples shows that indeed the exact same ones are missing. Instead of adding 28 largely correlated indicator variables, 12 indicator variables for groups were added (see Table 5).

The conservative approach to missing value imputation is also in line with the work by (Radcliffe & Surry, 2011, p. 27). They report that before variable selection, achieving reproducible results was very difficult. Having conservative imputation variables might alleviate overfitting.

Table 4: Result of the variable selection process. In total 18 variables out of 47 were selected into the reduced dataset.

Both Runs	**First Run**
HadBasketAdd	Dummy_PRODUCT_SINCE
Dummy_BASKET	VisitCountToday
ScreenWidth	InitBasketNonEmpty
PreviousVisitCount	PageIs(pageType = SEARCH)
ViewsOn(pageType = PRODUCT)	**Second Run**
ViewsOn(pageType = OVERVIEW)	<u>Benefiting Y_4:</u>
ViewCount	PageIs(pageType = OVERVIEW)
Dummy_SEARCH	BasketQuantity
WasConvertedBefore	<u>*Benefiting G*:</u>
Dummy_CONVERSION	VisibleTime(lastPageCount = 3)
NumberOfDifferent(pageType = PRODUCT)	PageIs(pageType = PRODUCT)
Dummy_PRODUCT	
Dummy_VISIT	
Dummy_TAB	

4.2.2 Variable Selection

Multiple papers suggest that feature selection is important to obtain reproducable results. Reasons for this are that the uplift effect is usually relatively small and an attempt to model it has a high risk of modeling much of the noise. (Lo, 2002; Radcliffe & Surry, 2011; Michel, et al., 2013)

For the reduced dataset variables were selected based on the variable importance a logistic regression. To account for the diversity of approaches that are tested in this work, three of the targeted variables were chosen: Y, Y^* and Y_4. For the Y_4 target variable the one-vs-rest (OvR) approach has been chosen. This method treats each of the four possible values of Y_4 as an independent class. Effectively four logistic regressions are trained. Thus, the total number of variable importances is six. Each of the six models votes on all variables with the highest relevance until 80% of the sum of the variable scores is reached. Finally,

all features with less than five votes are removed from the list. This way 18 features were selected out of the total of 47 for the reduced dataset. The selected variables and the votes they received are listed in Table 4.

The purpose of considering all used target variables in the feature selection process and only keeping generally popular variables after voting is to find variables that are relevant for all models. The drawback of this strategy is that a variable that is highly important to both the logistic regression and random forest classifier for one target might only get two votes and be dropped from the dataset. By being strict in this regard variables that are generally discriminative of the samples instead of highly correlated with the target are selected and should reduce the risk of overfitting.

The experiment was done for both the full and the reduced dataset. This allows to assess how sensitive uplift models are to disturbances in the features and how severe the risk of overfitting is. As will be discussed later, the results on the reduced dataset suggested that models relying on differentiating the treatment and control group and the four classes variable respectively perform poorly with this feature selection. A second variable selection was done with a tuned version of the algorithm described above.

The group indicator variable G is included in the set of target variables. Additionally, a random forest was used in addition to the logistic regression classifier. The total number of feature importance values from all 6 models is 11. The reason for this lies in reason the random forest classifier determines the importance. The feature importance is not a weight that is associated with one classifier in an OvR model. Instead the impact of a feature is measured on the out-of-bag (OOB) data that is associated with each tree in the tree. On those OOB datasets the accuracy is calculated once with the feature unaltered and once with the randomized feature. The difference between the two outcomes across all trees in the forest is the feature accuracy. Hence there is only one score even for a mutli-class target variable.

With the 11 importance scores for each feature the voting process that was described above was started again. The cutoff until which each model votes on the feature was chosen to be 50% of the sum over all

importance values and all features with two or more votes were included. By reducing the number of votes per model and in turn reducing the number of votes required more specialized features were chosen, e.g. features that are only used by the newly included group variable G. This way 18 features were selected like before.

The resulting set of features is largely like the one from the first process except for four features. Two of the dropped variables were not voted for at all in the second run. The other two variables that were dropped were voted for only by one model. The variables that were selected instead are more diverse, in that they're voted to not only by more than one model, but by models with different target variables. All the selected features and the difference to the first set are displayed in Table 4.

4.3 Selected models

The focus of the analysis is on the response and uplift probability approaches from Chapter 2. All the approaches can be implemented using any classification algorithm. Because there is no obvious choice, a set of viable classification algorithms have been selected for evaluation with the different approaches. Each will be introduced briefly and the relevant tunable parameters will be explained.

When comparing multiple algorithms, the parameters play an important role, in that they can greatly influence the generated model performance. Not tuning the parameters for each model would make a comparison of the algorithms impossible and would favor algorithms that are more robust. Luckily, many algorithms have just one or two tunable parameters.

Table 5: Overview of missing values and the imputation through indicator variables.

Variable Name	# missing	% missing	Indicators
TimeToFirst(pageType = CART)	2901978	95%	CART
TimeSinceFirst(pageType = CART)	2901978	95%	
TimeSinceOn(pageType = SALE)	2797914	92%	SALE_SINCE
TimeToFirst(pageType = SALE)	2776506	91%	SALE
TimeSinceFirst(pageType = SALE)	2776506	91%	
TimeOn(pageType = SALE)	2776506	91%	
TimeToBasketAdd	2758911	90%	BASKET
TimeSinceLastConversion	2760159	90%	
TimeSinceOn(pageType = SEARCH)	2318199	76%	SEARCH_SINCE
TimeToFirst(pageType = SEARCH)	2257875	74%	SEARCH
TimeSinceFirst(pageType = SEARCH)	2257875	74%	
TimeOn(pageType = SEARCH)	2257875	74%	
TimeSinceTabSwitch	1824336	60%	TAB
ViewCountLastVisit	1587621	52%	VISIT
TimeSinceLastVisit	1587621	52%	
TimeSinceFirstVisit	1587621	52%	
DurationLastVisit	1587621	52%	
TimeSinceOn(pageType = PRODUCT)	1108740	36%	PRODUCT_SINCE
TimeToFirst(pageType = PRODUCT)	894939	29%	PRODUCT
TimeSinceFirst(pageType = PRODUCT)	894939	29%	
TimeOn(pageType = PRODUCT)	894939	29%	
TabSwitchPer(pageType = PRODUCT)	894939	29%	
ClicksPer(pageType = PRODUCT)	894939	29%	
TimeSinceOn(pageType = OVERVIEW)	730275	24%	OVERVIEW_SINCE
TimeToFirst(pageType = OVERVIEW)	654318	21%	OVERVIEW
TimeSinceFirst(pageType = OVERVIEW)	654318	21%	
TimeOn(pageType = OVERVIEW)	654318	21%	
ScrollHeightPer(pageType = OVERVIEW)	654318	21%	

The logistic regression and linear SVM (Cortes & Vapnik, 1995) classifier are both regularized with a regularization parameter. The objective function includes a term with the norm of the model weights, which is scaled by this regularization parameter. Thus, a larger factor will increase the pressure towards smaller weights, which will help with preventing overfitting of the model. The other option that can be tuned is the norm that is used in the penalization. The L1 norm will lead to spare weights by forcing more towards zero. The result is a model that performs feature selection. The L2 norm is the default and increases the penalty with the square of the weight.

$$\min_{w,c} \|w\|_{L1/L2} + C \sum_{i=1}^{n} \log\left(\exp\left(-y_i(X_i^T w + x)\right) + 1\right) \tag{29}$$

Support vector machines (Cortes & Vapnik, 1995) work by finding a separating hyperplane that maximizes the margin between the closest training samples on either side of the plane. This minimizes the generalization error of the classifier. The optimization problem was solved in the primal form as this can lead to faster training times on large training datasets as this one, when the number of training samples is greater than the number feature dimensions (Chapelle, 2007). The SVM doesn't give meaningful probability scores by default. This can be overcome by using Platt scaling. This method fits a sigmoid function to the outputs of the SVM. To avoid overfitting the sigmoid is learned in a 4-fold cross-validation scheme, and the output averaged. (Platt & others, 1999) Besides the sigmoid function an isotonic regression model is tested as an alternative calibration method. (Chakravarti, 1989)

$$\min_{\omega,b,\zeta} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n} \zeta_i \tag{30}$$

subject to

$$y_i(\omega^T \phi(x_i) + b) \geq 1 - \zeta_i \qquad \forall i = 1 \dots n$$

$$\zeta_i \geq 0 \qquad \forall i = 1 \dots n$$

Random forests are constructed by growing multiple trees in a randomized fashion. The randomization helps make each tree be

different by forcing each to explore different splits of the dataset. This helps regularize the model, while giving good accuracy through voting among the many trees. To begin with, each tree is grown on a different dataset. Each tree gets a resampled version of the dataset where samples were drawn with replacement. At each branching point a split of the dataset is chosen. Instead of selecting the best split across all features and their best splitting point, the tree only gets a small subset of the features and their best splitting point to choose from. Extremely randomized forests on the other hand, also get a subset of the features at each split, but instead of the optimal split for each feature, they get a random split. Optimization hence only happens by selecting the best among several random splits.

Random forests (Breiman, 2001) and extremely randomized (Geurts, et al., 2006) trees have many parameters, but two were identified to be especially sensitive. The number of features that the algorithm chooses from at each split, and the minimal number of samples in each leave node. Increasing the number of features that are available for optimization increases the likelihood that features that can be exploited together in subsequent split can be exploited. On the other hand, making this value to high will reduce the variance in the forest too much, and will ultimately lead to overfitting of the model. Similarly controlling the minimal number of samples in the leave nodes prevents splits that focus on an overly small subset of the population. Other parameters include the number trees, the maximum depth, the maximum number of leaf nodes. However, by controlling the minimum samples per leaf the depth and maximum number of leaf nodes is also limited. The number of trees was chosen to be 300 as this is a good compromise between computational complexity and model robustness. For the extremely randomized trees the number of trees was set ten times as high to 3000. This is possible because the algorithm works much faster when not having to compute the optimal split for each chosen feature at each split.

The Gaussian naïve Bayes is a restrictive generative classifier that generally results in a quadratic decision boundary, unless the covariance is shared between the target classes. In the latter case the naïve Bayes classifier will resemble logistic regression classifier. The algorithm is based on the Bayes' formula for calculating conditional probabilities. It is

assumed – just as in logistic regression – that the features are conditionally independent of each other given the target. But unlike logistic regression the naïve Bayes classifier is tied to this assumption. In logistic regression, the weights are chosen arbitrarily instead of computing the empirical probabilities from the data. Hence, logistic regression is less restrictive than naïve Bayes and is more likely to over fit with few training samples. Even though the dataset in this work has many samples Gaussian naïve Bayes will be considered, because the training times are short. (Schapire, et al., 2007)

The k-nearest-neighbor classifier classifies new samples by finding the closest samples from the training set in the features space. The predicted class label is the most likely class among the neighbors or the likelihood of each class as estimated from the neighbors. The k specifies the number of closest samples to consider during prediction. Theoretically the model doesn't require any training, but to speed up the search for close samples an acceleration structure such as a search tree is constructed during training. The other tunable parameter is the weighting of the samples during prediction. Instead of weighting each sample equally, samples can be weighted by the inverse of their distance to the input sample.

4.4 Evaluation Strategy

To make the evaluation more robust and get an estimate of the model variability with changes to the training and test data, one can perform a k-fold cross-validation. Normally, the dataset is split into k parts and each part serves as the test set once, while the other parts are used for training. This way the whole dataset is used. In comparison when just splitting into a training and test set – for example 80% for training, 20% for testing – 20% of the data is never used for training.

In this case, the dataset is timestamped and the same customer can visit the store multiple times during the period. It is also possible that a customer removed the tracking cookie. Hence splitting the dataset as described above is not an option as data that temporally lies after the test data would be used for training.

Table 6: Overview of classification algorithms and their tuned parameters.

Logistic Regression estimates probabilities using a multinomial linear regression in a logistic function. Regularization penalizes the norm of the weight while minimizing the mean squared error.	Regularization scaling: $C = [10^{-8}, \dots, 10^{8}]$ Regularization norm: $L = [L1, L2]$
Calibrated Linear SVM constructs a separating hyperplane that maximizes the margin to the nearest samples on either side. Larger margins should give better generalization. For speed liblinear was used instead of libsvm.	Regularization factor: $C = [10^{-8}, \dots, 10^{8}]$ Calibration method: $M = [sigmoid, isotonic]$
Gaussian Naïve Bayes is a restrictive generative model based on Bayes' rule. It assumes that the variables are conditionally independent of each other given the target.	
Random Forest trains a forest of trees of multiple decision trees. The grown to be diverse through bagging and random sub-space selection to minimize variance and increase precision.	Minimum number of samples per leaf: $S_{min} = [100,200,1000,2000]$ Features considered per split: $F_{split} = [5, \dots, 12]$
Extremely Randomized Trees are constructed like random forests, but the trees are even more randomized. Instead of a of optimal splits from a random sub-space, random splits are considered at each split.	Minimum number of samples per leaf: $S_{min} = [100,200,1000,2000]$ Features considered per split: $F_{split} = [5, \dots, 12]$
KNN classifies samples by computing the majority class of similar training samples in the feature space. Similarity can be defined by any distance metric.	Number of neighbors: $K = [1,5,10,100,150, \dots$ $500,1000, \dots 4000]$

Other strategies include:

- Cross validation by moving a training and test set window through the time series
- Drawing k-1 new datasets from the original dataset with replacement

The advantage of the first method is, that the dataset is not altered in any way. But firstly, one needs enough data that over a long enough period to conduct this test. Secondly, at any step during the process not all the data is used for training and testing.

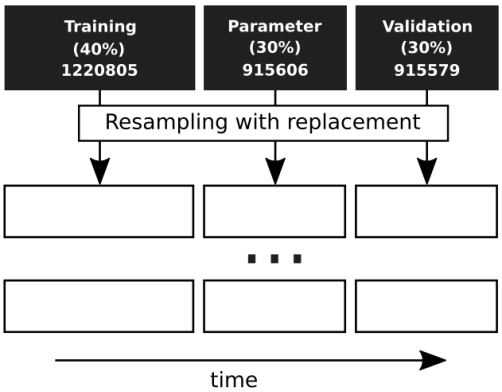

Figure 9: Cross-valuation strategy by resampling the original datasets with replacement.

The second method effectively creates new datasets by resampling the original dataset. Because samples are drawn with replacement, the new datasets will contain some samples multiple times, while other samples are not included at all. This makes the datasets sufficient different, to estimate the variability of the approaches and classifiers. It should be noted, that the dataset has enough samples that reducing the training and testing set for the sliding window approach would probably not drastically change the performance.

To prevent overestimating model performance through parameter tuning, a validation test was also introduced. All approaches with all classifiers and parameter combinations were trained on the training set. Then parameter selection was done for every classifier *for every approach*. Then those models were trained on the training and testing sets and evaluated on the validation set for the final assessment.

Table 7: Training, test, and validation set sizes.

Set	**#-samples**
Training	1220805 (40%)
Parameter	915606 (30%)
Validation	915579 (30%)

The complete process is as follows:

1. All models are trained on the training set and predict on the parameter-tuning set.
2. For all models the *percentual* qini curve and the accompanying qini score are calculated. This way both are automatically normalized to the size of the dataset, making a comparison between the first and second holdout set straight forward.
3. The best parameters per approach and classifier are selected based on the qini score *alone*.
4. The remaining models are trained on the training and parameter-tuning set, and the scores are predicted for the validation set.
5. The qini curves and scores are calculated for the remaining models
6. The qini curves are analysed using the recommendations by (Radcliffe & Surry, 2011; Kane, et al., 2014)

5 Results

All results in this chapter are the average over all 10 cross-validation datasets. For readability, the complete tables with the qini scores for all approaches and classifiers on the validation set, and the figures with the qini curves for the parameter-tuning and validation set can be found in "Appendix A: Uplift Score Tables" and "Appendix B: Qini Curves Plots" respectively. The abbreviations that are used for the models are summarized in Table 8.

Table 8: Abbreviations for uplift model approaches.

Response Model	Response
Naïve Two Model	NTM
Naïve One Model	NOM
Generalized Class Variable Transformation	Kane
Class Variable Transformation	CVT
Interaction Term Approach	ITA
Reflective Uplift Approach	Reflective
Pessimistic Uplift Approach	Pessimistic

5.1 Full Dataset Discussion

On the first validation set of the full dataset the selected models achieve mostly good results. This is not surprising, as the models were selected based on their score on this set. The results on the second holdout set reveal that this might be due to overfitting as the results are generally worse.

The response model and NTM approaches give solid results on the parameter-tuning set. For both, the extremely randomized trees and the KNN classifiers perform best with qini scores above 0.0055. Looking at the qini curves reveals some differences. The tree-based classifiers and logistic regression show significantly more uplift on the first quantiles than the KNN classifier. However, the results on the holdout set are consistently lower than on the tuning-set. The response with logistic regression and random forests still show solid but lower performance.

The ITA performs well with the logistic regression and SVM classifiers on the parameter-tuning set, but with increased variance compared to other approaches across cross-validation runs. The logistic regression achieves

a qini score of 0.0648 with a variance of 0.1340, while the variance of other approaches is below 0.0030. This can be an indicator, that the results are not be reproducible on the validation set, and this is in fact the case. But also, the SVM performs poorly, and regresses from a score of 0.0080 with slightly increased variance of 0.0048 to a negative qini score of -0.0010.

The CVT shows very solid performance on the parameter-tuning set with high scores and low variance across all cross-validations. On the validation set the performance is considerably worse, which again indicates that the model is overfitting the data. The reflective uplift approach doesn't perform well on both holdout sets. A reason for this might be that the classifiers are not able to distinguish between the two groups as they're too similar. This will be analyzed in Section 5.4.

The pessimistic uplift approach works well and performs as good as the CVT. The qini curves are much closer to the ones of CVT than the ones of the reflective uplift approach. This suggests that scores of the CVT dominate. A possible explanation is that since the reflective approach is not able to differentiate the classes all the variance in the predicted scores comes from the CVT. The analysis in Section 5.4 supports this (compare Figure 14 and Figure 15). An approach where a weighting parameter is optimized during parameter selection might improve results, though an improved reflective model (through further data pre-processing, better model tuning, etc.) is probably needed before having success with this approach. The results on the hold out are again worse than on the parameter-tuning set.

5.2 First Reduced Dataset Discussion

The reduced dataset generally performs better than the full dataset. Most importantly the results are much more consistent within approaches and between the parameter-tuning set and the validation set. This indicates that the variable selection processed outlined earlier (Chapter 4.2.2) is working as intended. The method selects variables that are specialized for the target variables that were used in the features selection process. This is evident from the results as the three models (Kane, NOM, Reflective) that also rely on differentiating between the treatment and control group perform worse than could be expected. Including one or

two more tests with the group variable as a target could alleviate the issue, and the results on the second feature selection run are presented in the next section.

Starting with the response model, we see that it performs very well on both holdout sets. The results are very repeatable on both sets, and the performance on the validation set is better. The best classifier is the random forest with 0.0058 and 0.0060 on the tuning and validation set respectively. The variance is low in both cases with 0.0021 and 0.0015. The NTM approach performs well on the validation set, but all classifiers give negative qini score on the validation set. As explained by (Hansotia & Rukstales, 2002) the interaction between the two classifiers and their individual prediction errors is likely the root cause. The results on the tuning set are very likely due to the parameter selection which has a high probability of finding at least one model per classifier that performs well on that test dataset.

The ITM again shows very similar results on the tuning set compared to the full dataset, though the extreme results in combination with the logistic regression are much reduced with matching reduction in variance. While the results on the validation set are less disastrous, but they're still much worse than on the tuning set. The improved generalization performance indicates that the explanation for the high variance on the full dataset was due to overfitting. This seems to still be the case in a reduced form.

The CVT and reflective uplift approach are again discussed together. The CVT performs like the response approach and shows repeatable results on the first and second holdout set. The reflective uplift approach doesn't perform well. As mentioned in the beginning of this section this is likely to the poor variable selection for this approach. While it does show positive results on the first holdout set, this is probably due to the parameter selection, and not accountable to generalization performance. The lackluster performance of the latter model should decrease the goodness of the pessimistic uplift approach, though as we saw on the full dataset, the impact of the reflective approach on the pessimistic approach might be small. This is indeed the case. The pessimistic uplift approach performs very like the CVT with qini scores that are a little lower for all classification algorithms. The fact that the CVT almost

mirrors the response model with slightly better performance, indicates that the dominant factor is the TR group. Under sampling of the CNR group or oversampling of the CR group as suggested by (Lai, et al., 2006) could change this characteristic.

Finally, the one model approaches show similar behavior with the approach by Kane consistently outperforming the NOM. The qini scores on the parameter-tuning set are among the best. Both get qini scores above 0.0080, but the variance is higher than for the CVT and response model approaches. As before this is a good indicator for the performance on the holdout dataset, and both significantly under performed. The approach by Kane still performs much better and the combination with the logistic regression shows a positive result. Due to the variance in the other classifiers this should be taken carefully, as this might not be repeatable.

5.3 Second Reduced Dataset Discussion

As described in Chapter 4.2.2, the second reduction run was done to address the poor performance of the models working on the Y_4 and G variables.

The reflective uplift model depends on the G variable and still doesn't perform well compared to other approaches. It did improve significantly over the first set of selected features. The average qini score improved from -0.0066 to -0.0035. The SVM and naïve Bayes gained, while the logistic regression stayed the same.

The pessimistic uplift approach decreased slightly which can be attributed to regressions in the CVT performance. The mean score of all three classifiers tested on the second feature selection were lower compared to those of the previous selection for CVT. The mean Qini score dropped from 0.0055 to 0.0046. The scores dropped equally across all the classification algorithms, which indicates that the results are stable and reproducible, and that the issue is not with one algorithm, but with the new set of features performing worse overall.

The models depending on the Y_4 target variable showed mixed results. On the one hand, the approach by Kane did not gain from the second feature selection run. The performance of the naïve Bayes and logistic

regression classifiers dropped compared to the first set of selected features. Only the SVM improved slightly but still scored a negative qini. On the other hand, the NOM approach gained significantly from the new set of features. The average qini score for the three tested classifiers improved from -0.0028 to 0.0018. Only the SVM decreased, while the logistic regression and naïve Bayes improved.

Finally, the response and naïve two model approach regressed as well, Since the CVT also decreased and was shown to perform like a response model it is likely the four features that were dropped in the new set of features in favor of the ones for the other target variables were discriminative of the conversion probability of customers.

Overall, the new set of features did improve the results of two of the models which were performing poorly, but there is still much potential in better feature selection. The regressions in the performance of other models indicates that including the features that were dropped in the second selection compared to the first, might bring them back up.

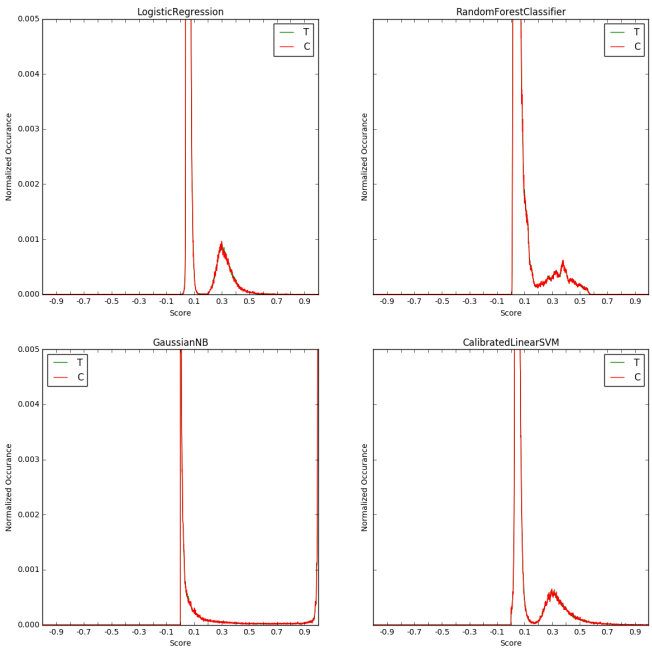

Figure 10: Score distribution of the response model

5.4 Detailed Analysis of Approaches

Response Model

The performance of the response model is without a doubt a surprise. The score distribution shows that the model can differentiate customers' response rate based on the given features (Figure 10).

A possible explanation for the good performance is that the response model can identify the responders and ranks them higher than non-responders. Of course, this happens in both the treatment and control group, because the responders in both groups are probably similar (and which can be seen from the similar score distributions in Figure 10). But because of the treatment there could be slight differences in their behavior. Since the response model is trained on the treatment group it'll more accurately model their behavior and concentrate converters a little better in the higher score ranges than the customers from the control group. Essentially, the responders in the control group will be "spread out" a little while the responders in the treatment group are "concentrated" a little more towards the higher deciles.

One indicator that this is the case can be found in the

Looking at the conversion rate in the deciles for treatment and control group separately, reveals that most of the conversions are happening in the top quantile. The response model can identify ~45% of the converters and gives them a high score in both groups. After that, the other half of the converters are spread out evenly across the deciles. However, almost 50% of the *differential* conversion rate is achieved in the decile after the "easily" identified converters, while there is a smaller -10% differential in the first decile (Figure 11). This supports the theory above, but more testing of this is needed to better understand the characteristics of the response model for uplift modeling. Especially, a test on a set of different datasets would show whether this is linked to some properties of the data. For instance, the response model could perform worse when there are more negative effects or if the heterogeneous treatment effect is larger.

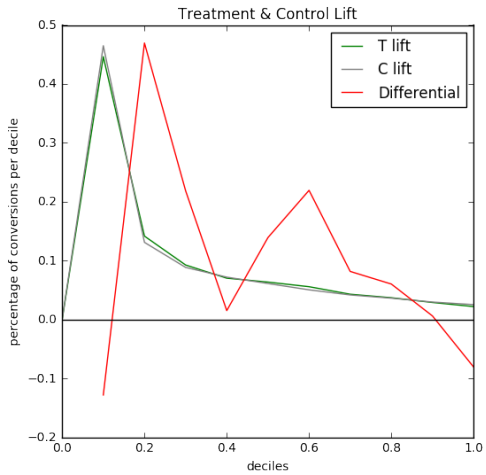

Figure 11: Lift per decile in percent of the total cumulative lift separated for treatment and control group.

Naïve Two Model Approach

The naïve two model approach performs poorly, but this cannot be attributed to the classifiers not being able to differentiate the converters from the non-converters as can be seen from the confusion matrices in Figure 12. The figure shows the confusion matrices for the treatment group, but the matrices for the control group classifier are the same, with virtually the same percentage of customers classified in the respective fields for each cutoff.

The two classifiers focus on predicting the main conversion effect and completely ignore the heterogeneous treatment effect. The score distribution further support this (Figure 12, left). The predicted uplift scores are all tightly clustered around zero and show no meaningful separation of customers.

Interaction Term Approach

The score distribution of the logistic regression classifier model in combination with ITA hint at possible explanations for the extreme qini curves of this model. As can be seen in Figure 13, the distribution of the output scores for the treatment and control group on the holdout dataset differ visibly, which was not the case for any other model that was tested.

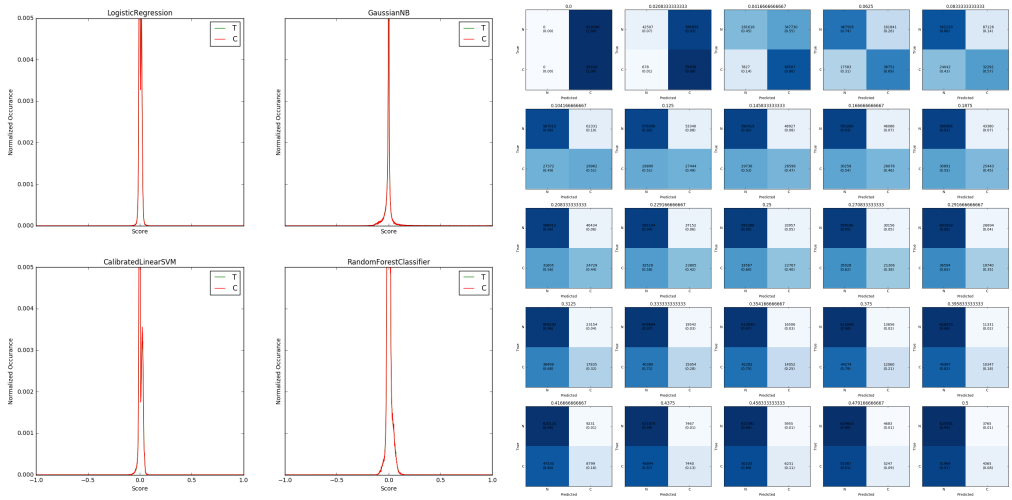

Figure 12: ***Left*** *– Score distribution of the naïve two model approach. The narrow bandwidth suggests a problem in differentiating samples.* ***Right*** *– Confusion matrices for the "treatment group" model (threashold range* [0.0, ... ,0.5]*).*

A model should not be able to put treatment and control customers in completely different score ranges. This is another observation that supports that the interaction term variables further exaggerate the overfitting issue that was seen for all models on the full dataset, such that it persists even on the reduced dataset.

The confusion matrices in Figure 13 show that the classifier can differentiate the converters from the non-converters as well as the response and NTM approaches. The image shows the case where the treatment indicator is set to $T = 1$, but with $T = 0$ the result was the same.

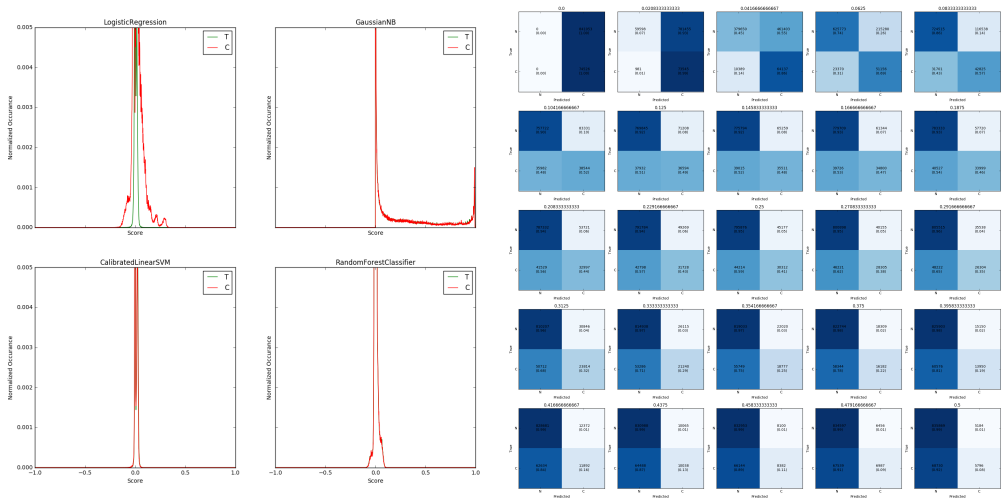

Figure 13: ***Left*** *– Score distribution for the ITA approach.* ***Right*** *– Confusion matrices for a logistic regression classifier and the interaction term set to* $T = 0$ *(threshold range* [0.2, ... ,0.6]*).*

Class Variable Transformation Approach

The CVT performs very well across all datasets, and the confusion matrices show that the classifier is indeed able to differentiate the transformed target variable (Figure 14). Comparing this to the confusion matrices of the ITA and response approach, the classification performance is worse. This is expected as the groups being combined into the two target classes are very dissimilar (one is converting and the other on is not), forcing the classifier to identify features that characterize the class but are not characteristic of converters or non-converters. At the same time the two target classes are identical in this regard, which makes classification difficult.

But being able to differentiate classes is barely the baseline requirement. Differentiating the right groups and ranking them by the right properties is what makes a good uplift model and the class variable transformation approach excels at that compared to other approaches. This shows in the score distribution plots, with all classification models mapping two groups of customers into different score ranges, but with relatively small differences, except for the NB classifier, which is over-confident because the independence assumption is violated.

That the two spikes in the score distribution chart are indeed the differentiation of the two class labels can be seen by computing the output score using the best threshold from the confusion matrices. In this case, this is approximately 0.3 and the positive and negative rate are 0.3 and 0.7 respectively. Plugging into equation (16) (Shaar, et al., 2016):

$$
\begin{aligned}
U_W(x) &= P(Y^* = 1|x) \cdot P(Y^* = 1) \\
&\quad - P(Y^* = 0|x) \cdot P(Y^* = 0) \\
&= 0.3 \cdot 0.3 - (1 - 0.3) \cdot 0.7 \\
&= 0.09 - 0.49 = -0.4
\end{aligned}
$$

This is the range of the first spike. As the threshold in the confusion matrix plot rises the customers are classified into differing classes, and while the two spikes are not even close to being purely from one class (as can be seen from the rather high error rates in the confusion matrix), there is non-the-less some correct differentiation. The good qini curve

results suggest that the slight differentiation is not only possible but that the class transformation is also sane. That is, the positive class more strongly correlates with an increased uplift than the negative class.

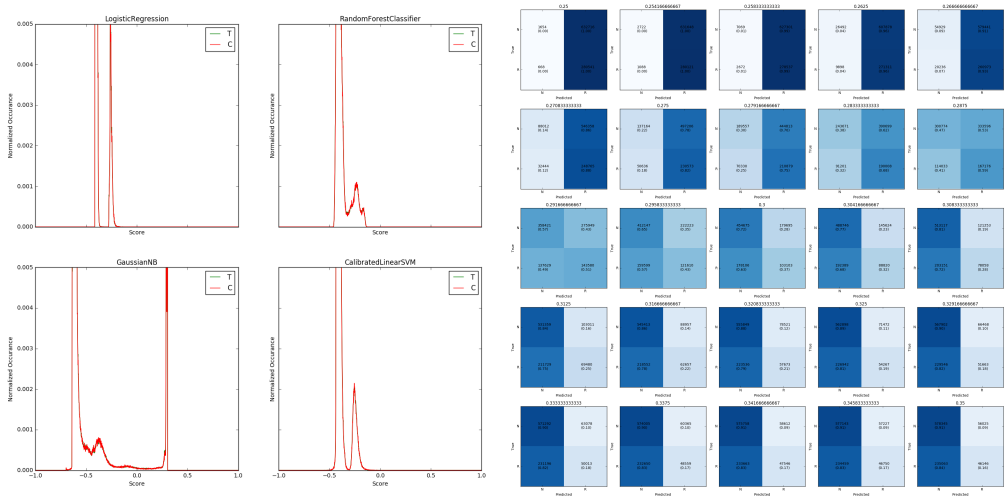

Figure 14: ***Left*** *– Score distribution for the CVT approach.* ***Right*** *– Confusion matrices for the CVT with a logistic regression. (threashold range* [0.25, … ,0.35]*).*

Reflective Uplift Approach

The reflective uplift model performs poorly on all three datasets. As it is the only model directly trying to differentiate the treatment from the control group, one possible explanation is that the two internal models are not able to differentiate between the two groups. A first support for this theory is given by the score distribution, which shows very little spread. This suggests that all samples get similar scores, because they're too similar. (see Figure 15) The confusion matrix supports this point. The model sharply classifies all samples into either the positive or negative class up to the critical cutoff. This is the range of scores into which all the samples are mapped, regardless of their target label. For this dataset, the task of predicting the group membership of a sample is very difficult.

Generalized CVT & Naïve One Model Approaches

Both models perform similarly, which is not unexpected. Since both models regressed when going from the full dataset to the reduced dataset, this could be explained with some important variables missing after selection.

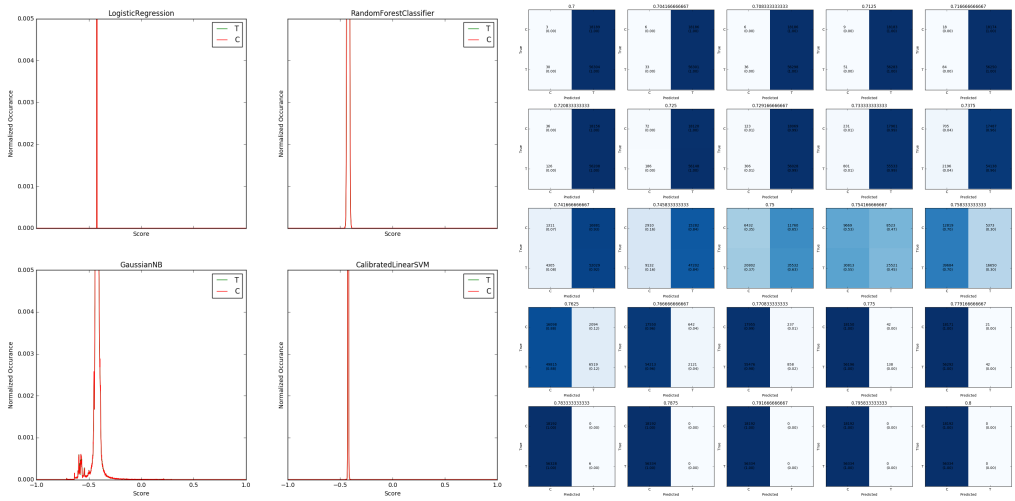

Figure 15: ***Left*** *– Score distribution of the reflective uplift model. The narrow bandwidth suggests a problem in differentiating samples.* ***Right*** *– Confusion matrices for the "conversion-group" classifier from the reflective uplift approach. (threashold range* [0.7, … ,0.8]*).*

Another possible explanation is that the two approaches could suffer from the class imbalance problem. The two converting classes are much less likely than the non-converting classes and the treatment class is overall more likely than the control class. The probability outputs reflect this, with mean probabilities of 0.69, 0.059, 0,23 and 0.019 for the TNR, TR, CNR and CR classes respectively. This is also the case for the CVT approach but the relative difference is much less severe with 0.31 for the positive class and 0.69 for the negative class.

5.5 Overall Analysis

The experiment shows that variable selection is essential to successful uplift modeling. The results on the full dataset are promising after parameter tuning, but the results on the validation set reveal that this is a case of overfitting. High variance in the outputs gives one parameter combination that performs well on the parameter-tuning set. In contrast the results on the first reduced dataset show that the models can in fact produce repeatable results across the parameter-tuning and validation set.

However, the generalized class variable transformation approach, the naïve one model approach, and the reflective uplift approach regressed with the reduced dataset, suggesting that some important features for those, were thrown out by the selection. Unfortunately, the second feature selection run did not show the intended improvements to the one model and reflective uplift approaches, even though the exchanged

features showed high relevance for the Y_4 and G target variables during selection. The reasons for this could be many, but the selection process as employed in this work has two possible shortcomings that when fixed might improve the performance of those approaches:

1. The classifiers for the G target variable were trained on the whole dataset during selection. In the reflective uplift model two models are trained, one for the converters and one for the non-converters. Reflecting this in the selection process, by using two classifiers could improve the selection.
2. The number of features that are selected is not optimized and chosen arbitrarily. There is no trivial solution to this problem, but a simple grid search with different number of selected features could suffice to further improve the results.

Of course, the poor performance can still be due the shortcomings approaches themselves as explained above.

Three models performed well across all datasets, and fulfill the criteria that were presented in Chapter 3.4 “Evaluation Strategy”. The response, CVT and pessimistic uplift approach all performed well and similar on the parameter-tuning sets and the validation sets, which is a good indicator that the models don’t over fit the data. Furthermore, all show low variance across the 10 cross-validation folds. Generally, the pessimistic approach has qini scores a little lower than the other two approaches which can be attributed to the poor performance of the reflective uplift approach.

Finally, the ITA also shows excellent performance with some classifiers. For example, the Naïve Bayes repeatedly performs well and shows low variance across cross-validation folds. But due to the anomalous results with the logistic regression classifier, that can be attributed to differing score distributions between treatment and control class, it is hard to recommend this approach based on the results in this work.

6 Conclusion

In this work, the indirect uplift modeling approaches for heterogeneous treatment effects with binary outcomes have been explored. After an introduction to the uplift modeling setting the indirect approaches are explained. The naïve one model approach is introduced which to the best knowledge of the author has not been tested in the literature before. For the analysis, the qini cures and scores have been explained in detail and some remarks regarding their interpretation have been given. It has been shown that feature selection is essential for successful uplift modeling and an approach for selecting features with the purpose of working across a diverse set of models has been developed. The selection strategy is based on the feature importance of multiple classifications models that are trained on the diverse set of target variables. Then a voting scheme is used to find the variables that rank highly among many models.

Based on a large real-world dataset, 8 approaches have been compared. The results show the class variable approach (CVT) and the simple response model perform well. The response model is a surprise as it doesn't consider any negative effects. This could be due to the classifier overfitting the treatment group slightly and hence giving those customer higher scores. But this is not definitive and more research is needed to explain the behavior that was seen. The pessimistic uplift approach is shown to be dominated by the CVT and overall its performance is slightly behind the CVT because the reflective uplift approach is not able to differentiate the treatment and control groups. Overall the pessimistic uplift approach still seems promising and with more work and further research the reflective uplift approach could still outperform the CVT.

The poor performance of the naïve two model approach points to the importance of post-processing as suggested by (Hansotia & Rukstales, 2002). Based on this idea, and the similarity to uplift trees which map properties onto uplift estimates of groups of samples, a two-stage model is proposed at the end of Chapter 2. The uplift model is a projection of customers into a one dimensional uplift space in which clustering can be done. The prediction is the empirical uplift of those clusters. This approach could be promising as the uplift trees have been shown to

perform very well, and could lead to a method with similar properties that doesn't rely on special algorithms.

Another field that needs more research is the impact of feature selection on uplift models. This work has shown that some form of selection is mandatory and that a selection can be found that works across several models. However, some models showed decreased performance after feature selection.

7 Bibliography

Breiman, L., 2001. Random forests. *Machine learning,* 45(1), pp. 5-32.

Chakravarti, N., 1989. Isotonic median regression: a linear programming approach. *Mathematics of operations research,* 14(2), pp. 303-308.

Chapelle, O., 2007. Training a support vector machine in the primal. *Neural computation,* 19(5), pp. 1155-1178.

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine learning,* 20(3), pp. 273-297.

Crone, S. F., Lessmann, S. & Stahlbock, R., 2006. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research,* 173(3), pp. 781-800.

Geurts, P., Ernst, D. & Wehenkel, L., 2006. Extremely randomized trees. *Machine learning,* 63(1), pp. 3-42.

Guelman, L., 2015. *Optimal personalized treatment learning models with insurance applications,* s.l.: s.n.

Guelman, L., Guillén, M. & Pérez-Marin, A. M., 2015. Uplift random forests. *Cybernetics and Systems,* Volume 46, pp. 230-248.

Hansotia, B. & Rukstales, B., 2002. Incremental value modeling. *Journal of Interactive Marketing,* 16(3), p. 35.

Jaroszewicz, S. & Zaniewicz, Ł., 2016. Székely Regularization for Uplift Modeling. In: *Challenges in Computational Statistics and Data Mining.* s.l.:Springer, pp. 135-154.

Jaskowski, M. & Jaroszewicz, S., 2012. *Uplift modeling for clinical trial data.* s.l., s.n.

Kane, K., Lo, V. S. Y. & Zheng, J., 2014. Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics,* 2(4), pp. 218-238.

Kang, P., Cho, S. & MacLachlan, D. L., 2012. Improved response modeling based on clustering, under-sampling, and ensemble. *Expert Systems with Applications,* 39(8), pp. 6738-6753.

Kim, D., Lee, H.-j. & Cho, S., 2008. Response modeling with support vector regression. *Expert Systems with Applications,* 34(2), pp. 1102-1108.

Kurt, W., 2015. *Bayesian A/B Testing: A Hypothesis Test that Makes Sense.* [Online]
Available at:
https://www.countbayesie.com/blog/2015/4/25/bayesian-ab-testing
[Accessed 3 3 2017].

Kuusisto, F. C., 2015. *Machine Learning for Medical Decision Support and Individualized Treatment Assignment,* s.l.: s.n.

Kuusisto, F. et al., 2014. *Support vector machines for differential prediction.* s.l., s.n., pp. 50-65.

Lai, L. Y.-T., 2006. *Influential marketing: A new direct marketing strategy addressing the existence of voluntary buyers,* Vancouver: s.n.

Lai, Y.-T.et al., 2006. *Direct marketing when there are voluntary buyers.* s.l., s.n., pp. 922-927.

Lo, V. S. Y., 2002. The true lift model: a novel data mining approach to response modeling in database marketing. *ACM SIGKDD Explorations Newsletter,* Volume 4, pp. 78-86.

Michel, R., Schnakenburg, I. & von Martens, T., 2013. Methods of variable pre-selection for net score modeling. *Journal of Research in Interactive Marketing,* 7(4), pp. 257-268.

Michel, R., Schnakenburg, I. & von Martens, T., 2014. A modified chi\^{} 22-test for uplift models with applications in marketing performance measurement. *arXiv preprint arXiv:1401.7001.*

Olson, D. L. & Chae, B. K., 2012. Direct marketing decision support through predictive customer response modeling. *Decision Support Systems,* 54(1), pp. 443-451.

Pechyony, D., Jones, R. & Li, X., 2013. A Joint Optimization of Incrementality and Revenue to Satisfy both Advertiser and Publisher. *Www,* pp. 123-124.

Platt, J. & others, 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers,* 10(3), pp. 61-74.

Radcliffe, N. J., 2007. Using control groups to target on predicted lift: Building and assessing uplift models. *Direct Market J Direct Market Assoc Anal Council,* Volume 1, pp. 14-21.

Radcliffe, N. J. & Surry, P. D., 2011. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions.*

Rokach, L., Naamani, L. & Shmilovici, A., 2008. Pessimistic cost-sensitive active learning of decision trees for profit maximizing targeting campaigns. *Data Mining and Knowledge Discovery,* Volume 17, pp. 283-316.

Rzepakowski, P. & Jaroszewicz, S., 2010. *Decision trees for uplift modeling.* Sydney, 2010 IEEE International Conference on Data Mining, pp. 441-450.

Rzepakowski, P. & Jaroszewicz, S., 2012. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems,* Volume 32, pp. 303-327.

Rzepakowski, P. & Jaroszewicz, S., 2012. Uplift modeling in direct marketing. *Journal of Telecommunications and Information Technology,* pp. 43-50.

Schapire, R., Blei, D. & Yoon, J. S., 2007. *COS 424: Interacting with Data.* Princeton(New Jersey): s.n.

Shaar, A., Abdessalem, T. & Segard, O., 2016. Pessimistic Uplift Modeling. *arXiv preprint arXiv:1603.09738.*

Shin, H. & Cho, S., 2006. Response modeling with support vector machines. *Expert Systems with applications,* 30(4), pp. 746-760.

Sołtys, M., Jaroszewicz, S. & Rzepakowski, P., 2015. Ensemble methods for uplift modeling. *Data mining and knowledge discovery,* Volume 29, pp. 1531-1559.

Stich, S., 2014. *Use the Beta Distribution.* [Online]
Available at:
http://sl8r000.github.io/ab_testing_statistics/use_the_beta_distribution/
[Accessed 24 2 2017].

Tang, Z., 2011. Improving direct marketing profitability with neural networks. *International Journal of Computer Applications,* 29(5), pp. 13-18.

Tian, L., Alizadeh, A. A., Gentles, A. J. & Tibshirani, R., 2014. A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association,* 109(508), pp. 1517-1532.

Wager, S. & Athey, S., 2015. Estimation and inference of heterogeneous treatment effects using random forests. *arXiv preprint arXiv:1510.04342.*

Weisberg, H. I. & Pontes, V. P., 2015. Post hoc subgroups in clinical trials: Anathema or analytics?. *Clinical Trials,* 12(4), pp. 357-364.

Winterberry Group & Direct Marketing Association, 2016. *statista.com.* [Online]
Available at: https://www.statista.com/statistics/289174/direct-mail-marketing-spending-us/
[Accessed 26 3 2017].

Zaniewicz, L. & Jaroszewicz, S., 2013. *Support vector machines for uplift modeling.* Dallas, 2013 IEEE 13th International Conference on Data Mining Workshops, pp. 131-138.

Appendix A: Uplift Score Tables

*Table 9: Qini scores and uplift percentage per quantile for all approaches on the second validation set of the **full** dataset.*

Approach		Q mean	Q std	0,0%	10,0%	20,0%	30,0%	40,0%	50,0%	60,0%	70,0%	80,0%	90%	100%
Response	**RF**	0,0044	0,0019	0,00%	-0,33%	**1,23%**	**1,80%**	**1,99%**	**2,14%**	2,21%	2,19%	2,34%	2,41%	2,57%
	LR	0,0038	0,0019	0,00%	-0,33%	1,13%	1,62%	1,54%	1,75%	1,93%	2,11%	2,15%	2,40%	2,34%
	SVM	0,0021	0,0018	0,00%	-0,12%	0,38%	0,74%	1,47%	2,07%	1,94%	2,20%	2,59%	2,44%	2,57%
	NB	-0,0013	0,0020	0,00%	-0,26%	-0,02%	0,29%	0,64%	1,19%	1,79%	2,14%	2,21%	2,33%	2,57%
NOM	**LR**	0,0055	0,0023	0,00%	**0,66%**	**1,29%**	**1,70%**	**1,83%**	2,07%	2,06%	2,17%	2,19%	2,28%	2,39%
	NB	0,0033	0,0033	0,00%	0,29%	0,68%	1,04%	1,43%	1,64%	2,08%	2,40%	**2,75%**	2,61%	2,57%
	RF	0,0021	0,0023	0,00%	0,46%	0,65%	0,89%	0,97%	1,33%	1,70%	2,06%	2,21%	2,46%	2,35%
	SVM	0,0021	0,0021	0,00%	0,04%	0,34%	0,95%	1,46%	2,00%	1,93%	2,18%	2,43%	2,39%	2,57%
Reflective	**NB**	0,0051	0,0033	0,00%	0,45%	0,90%	1,34%	1,58%	**2,10%**	**2,38%**	**2,56%**	2,63%	2,74%	2,57%
	SVM	0,0023	0,0024	0,00%	0,22%	0,32%	0,77%	1,04%	1,32%	1,76%	2,36%	**2,92%**	**3,20%**	**2,57%**
	RF	-0,0003	0,0016	0,00%	0,35%	0,52%	0,67%	0,89%	1,25%	1,59%	1,79%	1,94%	2,29%	2,57%
	LR	-0,0007	0,0014	0,00%	0,28%	0,47%	0,74%	1,05%	1,10%	1,38%	1,55%	2,04%	2,22%	2,57%
Pessimistic	**KNN**	-0,0012	0,0025	0,00%	-0,39%	0,36%	0,79%	0,94%	1,22%	1,46%	1,73%	1,93%	2,24%	2,56%
	LR	0,0076	0,0029	0,00%	**0,81%**	**1,32%**	**1,79%**	**2,11%**	**2,51%**	**2,71%**	**2,81%**	2,72%	2,43%	2,57%
	NB	0,0038	0,0020	0,00%	-0,34%	1,17%	1,50%	1,65%	1,86%	1,99%	2,02%	2,10%	2,27%	2,31%
	SVM	-0,0003	0,0018	0,00%	0,11%	0,08%	0,17%	0,84%	1,46%	1,74%	2,18%	2,31%	2,36%	2,57%
NTM	**LR**	-0,0004	0,0025	0,00%	0,10%	0,35%	0,69%	0,98%	1,25%	1,52%	1,82%	2,08%	2,21%	2,53%
	SVM	-0,0004	0,0018	0,00%	-0,54%	0,09%	0,38%	1,07%	1,42%	1,53%	2,16%	2,67%	2,38%	2,57%
	NB	0,0037	0,0020	0,00%	-0,16%	1,09%	1,41%	1,40%	1,59%	2,12%	2,17%	2,10%	2,29%	2,30%
ITA	**RF**	0,0031	0,0018	0,00%	0,23%	0,47%	0,78%	1,46%	2,00%	**2,30%**	2,45%	2,52%	2,48%	2,57%
	NB	-0,0007	0,0017	0,00%	-0,18%	-0,04%	0,27%	0,60%	1,34%	1,83%	2,21%	2,41%	2,46%	2,57%
	LR	-0,0008	0,0019	0,00%	0,24%	0,39%	0,63%	0,84%	1,20%	1,49%	1,75%	1,96%	2,25%	2,56%
	SVM	0,0015	0,0027	0,00%	0,63%	1,14%	1,52%	1,82%	1,98%	2,00%	2,07%	2,36%	**2,75%**	**3,29%**
CVT	**LR**	0,0010	0,0024	0,00%	0,15%	0,23%	0,38%	1,16%	1,50%	1,90%	2,35%	2,45%	2,43%	2,57%
	RF	-0,0003	0,0016	0,00%	0,21%	0,35%	0,65%	0,92%	1,27%	1,59%	1,89%	2,07%	2,29%	2,57%
	SVM	-0,0004	0,1036	0,00%	**1,83%**	0,46%	-4,26%	-9,03%	-14,13%	-20,88%	-28,12%	-31,52%	-32,68%	-30,64%
	NB	-0,0006	0,0019	0,00%	0,17%	0,33%	0,59%	0,91%	1,23%	1,53%	1,86%	2,08%	2,31%	2,57%
Kane	**LR**	-0,0010	0,0029	0,00%	0,12%	0,50%	0,79%	1,00%	1,32%	1,37%	1,49%	1,86%	2,18%	2,57%
	NB	0,0039	0,0018	0,00%	-0,32%	1,21%	1,55%	1,72%	1,69%	2,06%	2,13%	2,18%	2,28%	2,36%
	SVM	0,0029	0,0025	0,00%	-0,32%	0,73%	1,22%	1,68%	1,87%	2,12%	2,27%	2,43%	2,45%	2,57%

*Table 10: Qini scores and uplift percentage per quantile for all approaches on the second validation set of the **first reduced** dataset.*

Approach		Q mean	Q std	0,0%	10,0%	20,0%	30,0%	40,0%	50,0%	60,0%	70,0%	80,0%	90%	100%
Response	**RF**	**0,0060**	0,0015	0,00%	-0,39%	**1,70%**	1,84%	**2,14%**	2,15%	2,26%	2,49%	2,73%	2,61%	2,57%
	LR	0,0055	0,0019	0,00%	-0,49%	1,44%	**2,03%**	2,09%	1,98%	2,06%	**2,64%**	2,65%	**2,71%**	2,57%
	NB	0,0053	0,0020	0,00%	0,18%	0,82%	1,82%	1,93%	2,16%	**2,53%**	2,47%	2,41%	2,56%	2,57%
	ERT	0,0053	0,0020	0,00%	-0,52%	1,35%	**2,14%**	2,10%	**2,22%**	2,30%	2,48%	2,37%	2,43%	2,57%
	SVM	0,0052	0,0017	0,00%	-0,43%	1,37%	1,68%	**2,37%**	1,81%	2,04%	2,48%	**2,84%**	2,65%	2,57%
NOM	**LR**	-0,0015	0,0017	0,00%	-0,46%	0,20%	0,72%	1,27%	1,45%	1,55%	1,65%	1,72%	1,99%	2,57%
	RF	-0,0018	0,0025	0,00%	0,30%	0,53%	0,64%	0,80%	1,03%	1,21%	1,50%	1,78%	2,02%	2,57%
	SVM	-0,0027	0,0029	0,00%	-0,80%	0,32%	0,78%	0,96%	1,31%	1,34%	1,32%	1,54%	2,07%	2,57%
	NB	-0,0043	0,0043	0,00%	0,20%	0,45%	0,54%	0,73%	0,85%	1,05%	1,17%	0,95%	1,38%	**2,57%**
	ERT	-0,0048	0,0034	0,00%	**0,53%**	0,43%	0,49%	0,47%	0,44%	0,47%	0,87%	1,27%	1,81%	2,57%
Reflective	**ERT**	-0,0004	0,0035	0,00%	0,37%	0,54%	0,83%	0,85%	1,08%	1,31%	1,74%	2,14%	2,35%	2,57%
	RF	-0,0020	0,0017	0,00%	0,16%	0,25%	0,32%	0,55%	0,74%	1,16%	1,71%	2,19%	2,45%	**2,57%**
	LR	-0,0037	0,0019	0,00%	-0,51%	-0,15%	0,09%	0,48%	0,92%	1,29%	1,60%	1,91%	2,22%	2,57%
	SVM	-0,0067	0,0043	0,00%	-0,37%	-0,29%	-0,01%	0,09%	0,38%	0,77%	1,07%	1,44%	1,83%	**2,57%**
	NB	-0,0093	0,0037	0,00%	-0,44%	-0,35%	-0,18%	-0,05%	0,09%	0,24%	0,34%	1,03%	1,61%	2,57%
Pessimistic	**SVM**	0,0057	0,0017	0,00%	-0,49%	1,40%	**2,07%**	2,06%	2,17%	2,19%	2,54%	**2,76%**	2,57%	2,57%
	RF	0,0053	0,0021	0,00%	-0,40%	1,44%	1,83%	**2,15%**	2,12%	2,22%	2,43%	2,49%	2,63%	2,57%
	ERT	0,0047	0,0018	0,00%	-0,61%	**1,63%**	1,94%	1,99%	1,99%	2,10%	2,27%	2,43%	2,58%	2,57%
	NB	0,0044	0,0021	0,00%	-0,53%	0,74%	1,61%	1,73%	2,06%	2,36%	2,51%	2,72%	**2,80%**	2,57%
	LR	0,0035	0,0017	0,00%	-0,73%	0,74%	1,12%	1,54%	2,06%	**2,47%**	2,57%	2,63%	2,69%	2,57%
NTM	**RF**	-0,0002	0,0023	0,00%	0,32%	0,60%	0,94%	1,17%	1,23%	1,36%	1,63%	1,93%	2,18%	2,57%
	SVM	-0,0031	0,0020	0,00%	-0,36%	0,17%	0,69%	1,03%	1,28%	1,20%	1,23%	1,30%	1,96%	2,57%
	NB	-0,0043	0,0042	0,00%	0,19%	0,45%	0,54%	0,74%	0,85%	1,06%	1,16%	0,88%	1,38%	**2,57%**
	LR	-0,0071	0,0029	0,00%	-0,38%	0,03%	0,33%	0,46%	0,42%	0,61%	0,64%	0,87%	1,45%	2,57%
ITA	**LR**	**0,0248**	0,2027	0,00%	**7,40%**	**5,31%**	**4,26%**	**3,33%**	**2,54%**	2,09%	**2,61%**	**3,71%**	**5,12%**	2,57%
	NB	0,0055	0,0015	0,00%	0,03%	1,27%	1,77%	1,59%	1,96%	**2,40%**	**2,68%**	2,63%	**2,75%**	2,57%
	RF	0,0031	0,0021	0,00%	**0,62%**	1,18%	1,40%	1,50%	1,58%	1,88%	2,07%	2,18%	2,24%	2,57%
	SVM	-0,0022	0,0029	0,00%	-0,45%	0,62%	0,93%	1,21%	1,37%	1,35%	1,26%	1,35%	1,71%	2,57%
CVT	**SVM**	**0,0059**	0,0018	0,00%	-0,46%	1,48%	2,03%	**2,18%**	2,17%	2,19%	2,42%	**2,91%**	2,56%	2,57%
	RF	**0,0059**	0,0019	0,00%	-0,42%	**1,61%**	1,87%	1,96%	**2,19%**	**2,38%**	2,54%	2,68%	2,65%	2,57%
	LR	**0,0058**	0,0018	0,00%	-0,50%	1,48%	2,02%	2,03%	**2,22%**	2,15%	2,46%	**2,92%**	2,56%	2,57%
	ERT	0,0056	0,0016	0,00%	-0,64%	**1,52%**	**2,20%**	2,13%	2,01%	2,32%	2,36%	2,62%	2,67%	2,57%
	NB	0,0049	0,0021	0,00%	0,01%	0,71%	1,56%	1,67%	**2,20%**	**2,52%**	**2,65%**	2,41%	**2,69%**	2,57%
Kane	**ERT**	0,0054	0,0020	0,00%	0,12%	0,91%	1,71%	1,82%	2,01%	2,31%	**2,68%**	2,75%	2,69%	2,57%
	NB	0,0000	0,0034	0,00%	**0,38%**	0,35%	0,47%	0,85%	1,23%	1,74%	1,97%	2,26%	2,34%	2,57%
	LR	-0,0015	0,0035	0,00%	**0,47%**	0,65%	0,54%	0,70%	0,89%	1,14%	1,59%	1,82%	2,29%	2,57%
	RF	-0,0028	0,0038	0,00%	0,31%	0,41%	0,52%	0,74%	0,92%	1,01%	1,15%	1,75%	1,98%	**2,57%**
	ERT	-0,0091	0,0021	0,00%	-0,67%	-0,72%	-0,48%	-0,05%	0,30%	0,49%	0,86%	1,21%	1,55%	2,57%

*Table 11: Qini scores and uplift percentage per quantile for all approaches on the second validation set of the **second reduced** dataset.*

Approach		Q mean	Q std	0,0%	10,0%	20,0%	30,0%	40,0%	50,0%	60,0%	70,0%	80,0%	90%	100%
Response	**NB**	0,0048	0,0019	0,00%	-0,06%	0,97%	**1,91%**	2,05%	**2,14%**	1,92%	2,37%	2,43%	2,64%	2,57%
	SVM	0,0047	0,0019	0,00%	-0,35%	1,28%	1,76%	**2,29%**	1,96%	2,02%	2,36%	2,38%	2,56%	2,57%
	LR	0,0042	0,0020	0,00%	-0,30%	0,88%	1,50%	1,54%	1,88%	**2,23%**	**2,69%**	**2,67%**	2,70%	2,57%
NOM	**NB**	**0,0057**	0,0060	0,00%	**1,01%**	**1,63%**	1,75%	1,77%	1,67%	2,10%	2,26%	2,56%	2,58%	**2,57%**
	LR	**0,0056**	0,0021	0,00%	-0,04%	1,00%	1,47%	1,95%	2,05%	**2,44%**	**2,83%**	**2,66%**	**2,76%**	2,57%
	SVM	-0,0060	0,0025	0,00%	-0,43%	0,13%	0,48%	0,58%	0,72%	0,74%	0,63%	0,96%	1,79%	2,57%
Reflective	**NB**	-0,0015	0,0026	0,00%	**0,17%**	0,33%	0,46%	0,82%	0,84%	1,01%	1,71%	1,95%	**2,82%**	**2,57%**
	LR	-0,0037	0,0019	0,00%	-0,51%	-0,15%	0,09%	0,48%	0,92%	1,29%	1,60%	1,91%	2,22%	2,57%
	SVM	-0,0054	0,0019	0,00%	-0,20%	0,04%	0,22%	0,28%	0,60%	0,87%	1,10%	1,47%	1,84%	**2,57%**
Pessimistic	**SVM**	0,0053	0,0020	0,00%	-0,41%	1,37%	**1,85%**	**2,19%**	**2,22%**	2,13%	2,37%	**2,62%**	2,55%	2,57%
	LR	0,0045	0,0018	0,00%	-0,57%	0,72%	1,80%	2,01%	2,03%	**2,20%**	**2,62%**	**2,59%**	2,72%	2,57%
	NB	0,0044	0,0019	0,00%	-0,45%	0,72%	1,49%	2,02%	**2,31%**	**2,20%**	2,22%	2,56%	**2,89%**	2,57%
NTM	**NB**	**0,0057**	0,0060	0,00%	**1,01%**	**1,63%**	1,75%	1,77%	1,67%	2,10%	2,26%	2,55%	2,57%	**2,57%**
	RF	0,0001	0,0025	0,00%	0,01%	0,45%	0,77%	1,17%	1,53%	1,70%	1,85%	2,06%	2,16%	2,57%
	LR	-0,0014	0,0021	0,00%	0,12%	0,72%	0,91%	1,20%	1,39%	1,39%	1,53%	1,24%	1,73%	2,57%
	SVM	-0,0044	0,0019	0,00%	-0,43%	0,30%	0,58%	0,86%	0,84%	0,82%	0,94%	1,37%	1,88%	2,57%
ITA	**LR**	**0,0158**	0,1875	0,00%	**6,47%**	**4,44%**	**2,87%**	**2,18%**	2,04%	1,58%	1,88%	2,34%	**3,55%**	2,57%
	NB	**0,0062**	0,0015	0,00%	**0,48%**	**1,38%**	**2,38%**	**2,29%**	**2,54%**	1,91%	2,24%	2,13%	2,45%	2,57%
	SVM	0,0007	0,0023	0,00%	-0,08%	0,84%	1,14%	1,55%	1,77%	1,77%	1,63%	1,64%	2,02%	2,57%
CVT	**SVM**	0,0053	0,0020	0,00%	-0,40%	**1,39%**	**1,86%**	**2,19%**	**2,15%**	2,03%	**2,40%**	**2,60%**	2,67%	2,57%
	LR	0,0046	0,0018	0,00%	-0,56%	0,72%	1,80%	2,02%	2,03%	**2,21%**	**2,62%**	2,58%	2,72%	2,57%
	NB	0,0038	0,0019	0,00%	-0,34%	0,75%	1,65%	2,08%	2,07%	2,01%	1,93%	2,50%	**2,78%**	2,57%
Kane	**NB**	-0,0030	0,0027	0,00%	-0,03%	0,12%	0,28%	0,49%	0,60%	1,01%	1,32%	2,09%	2,67%	**2,57%**
	LR	-0,0070	0,0015	0,00%	-0,66%	-0,20%	-0,08%	0,12%	0,40%	0,74%	1,05%	1,42%	1,79%	2,57%
	SVM	-0,0075	0,0021	0,00%	-0,47%	-0,20%	-0,18%	0,09%	0,39%	0,63%	1,02%	1,26%	1,58%	2,57%

Appendix B: Qini Curves Plots

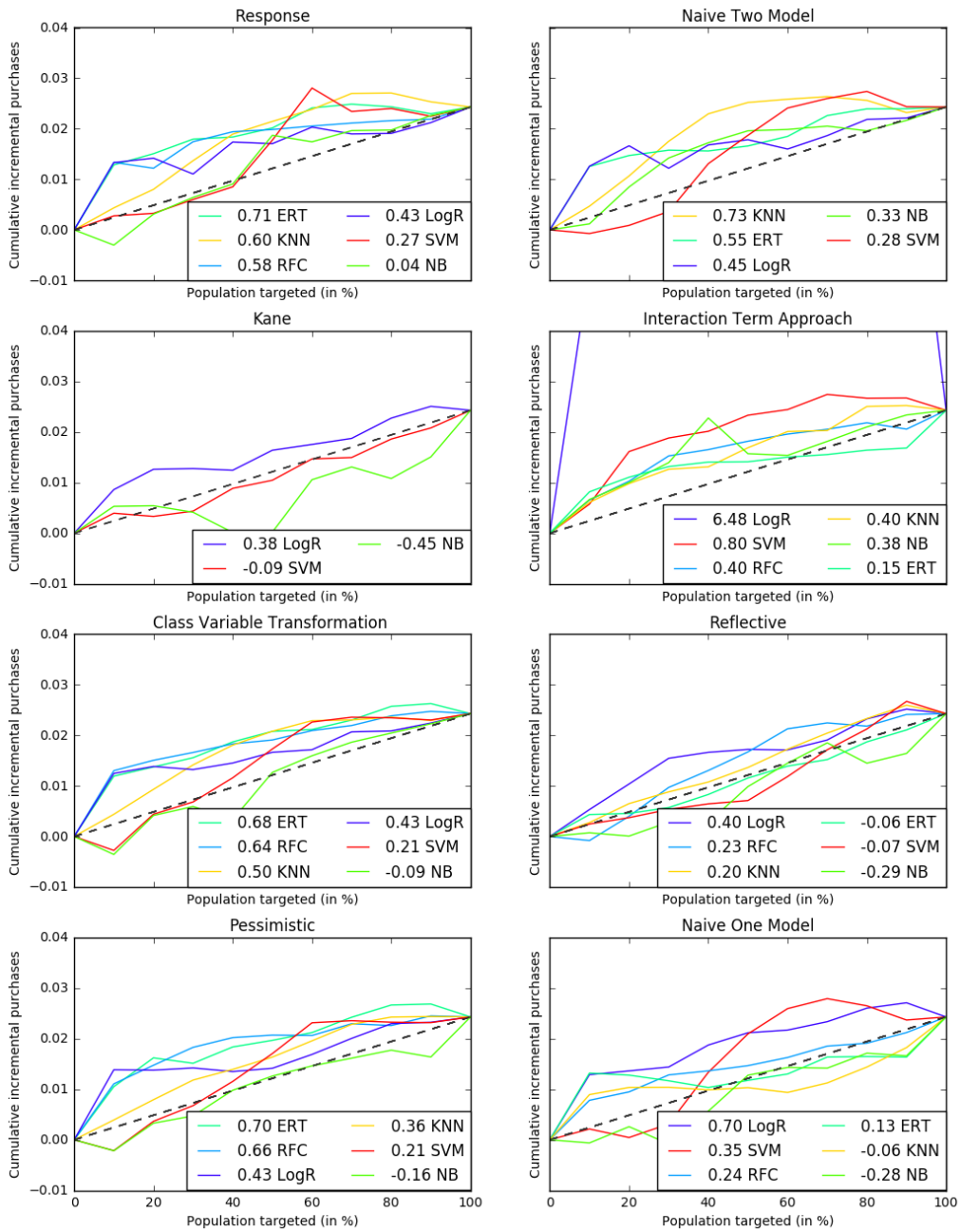

*Figure 16: Qini curves from the **first** validation set of the **full** dataset.*

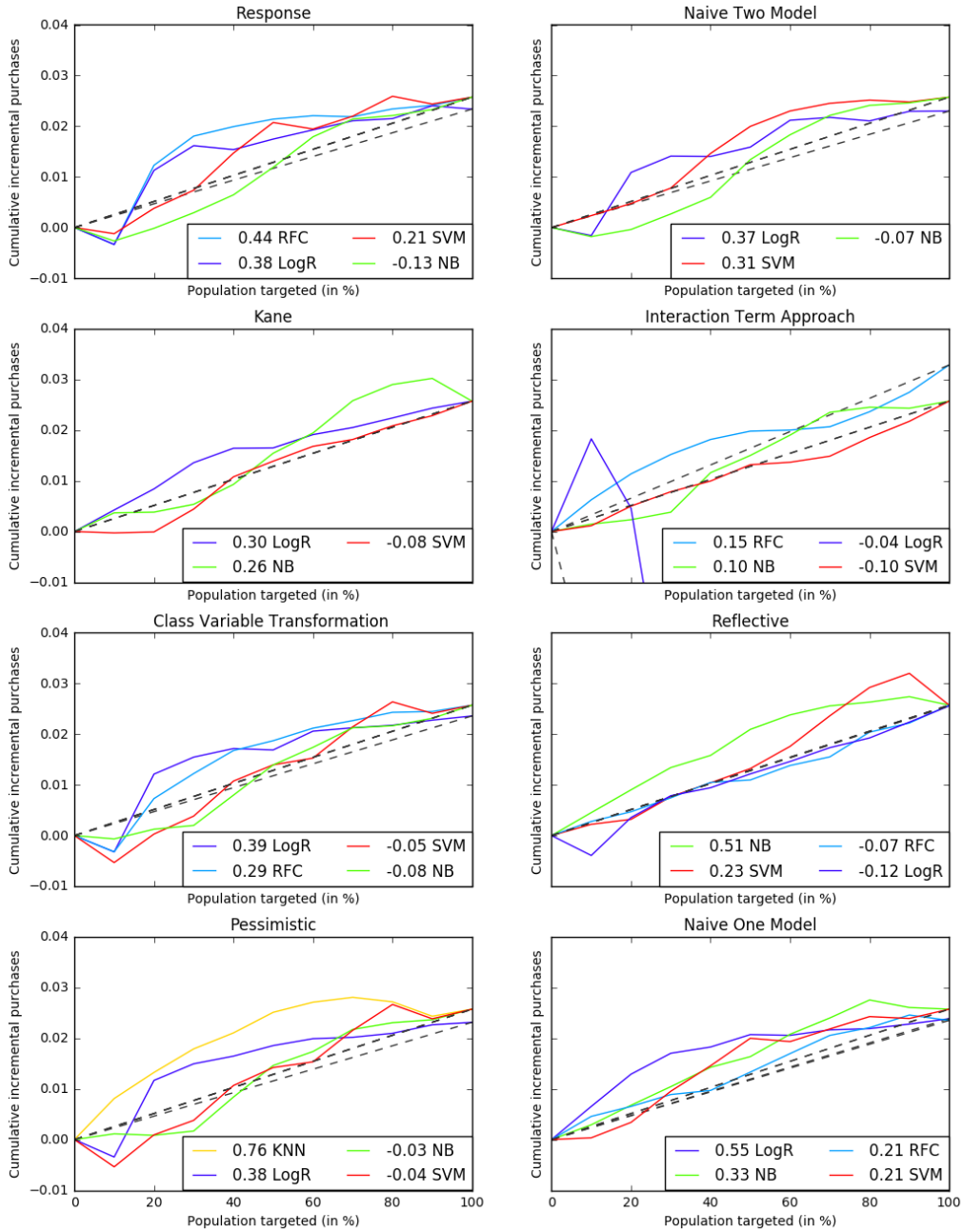

*Figure 17: Qini curves from the **second** validation set of the **full** dataset.*

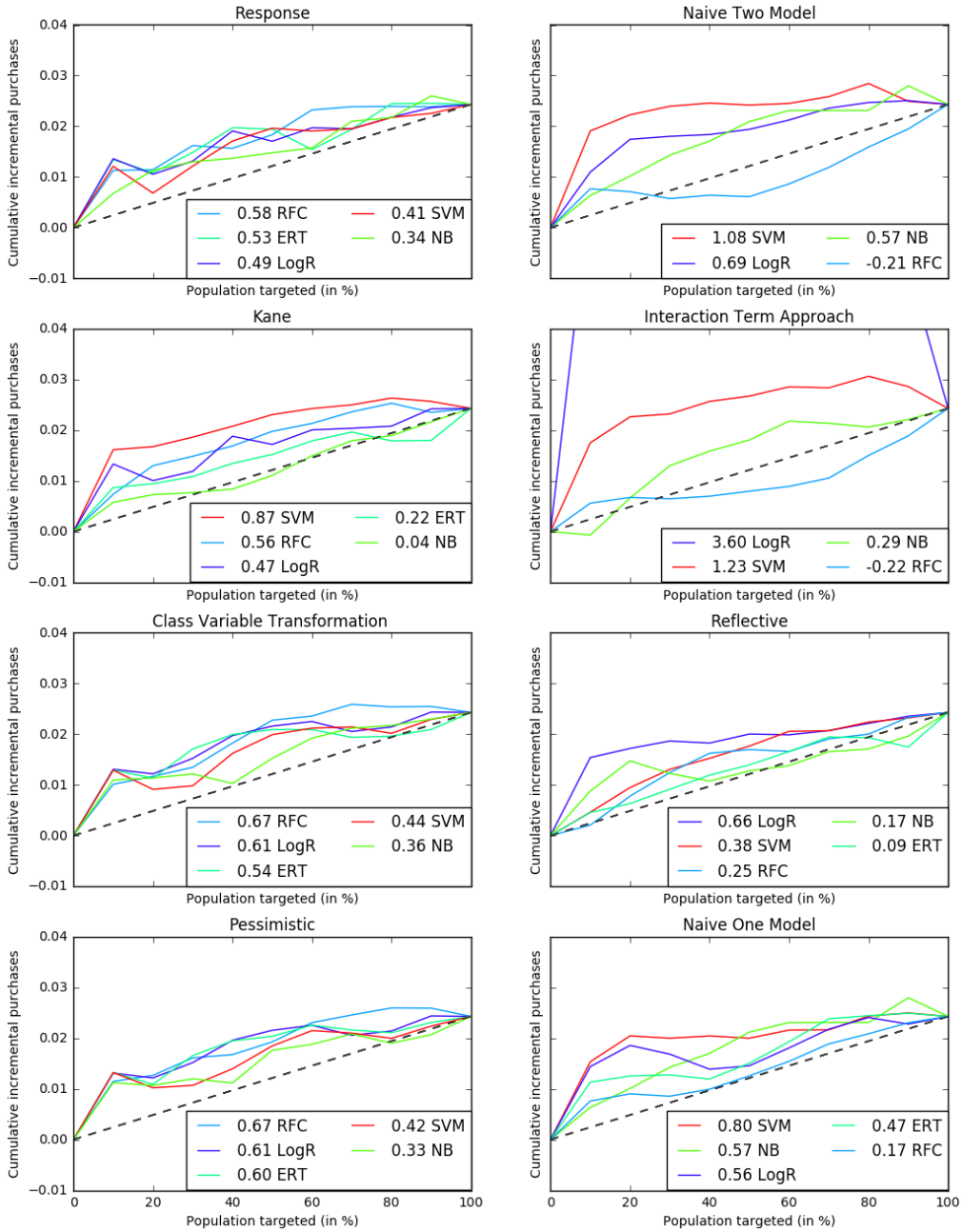

*Figure 18: Qini curves from the **first** validation set of the **first** reduced dataset.*

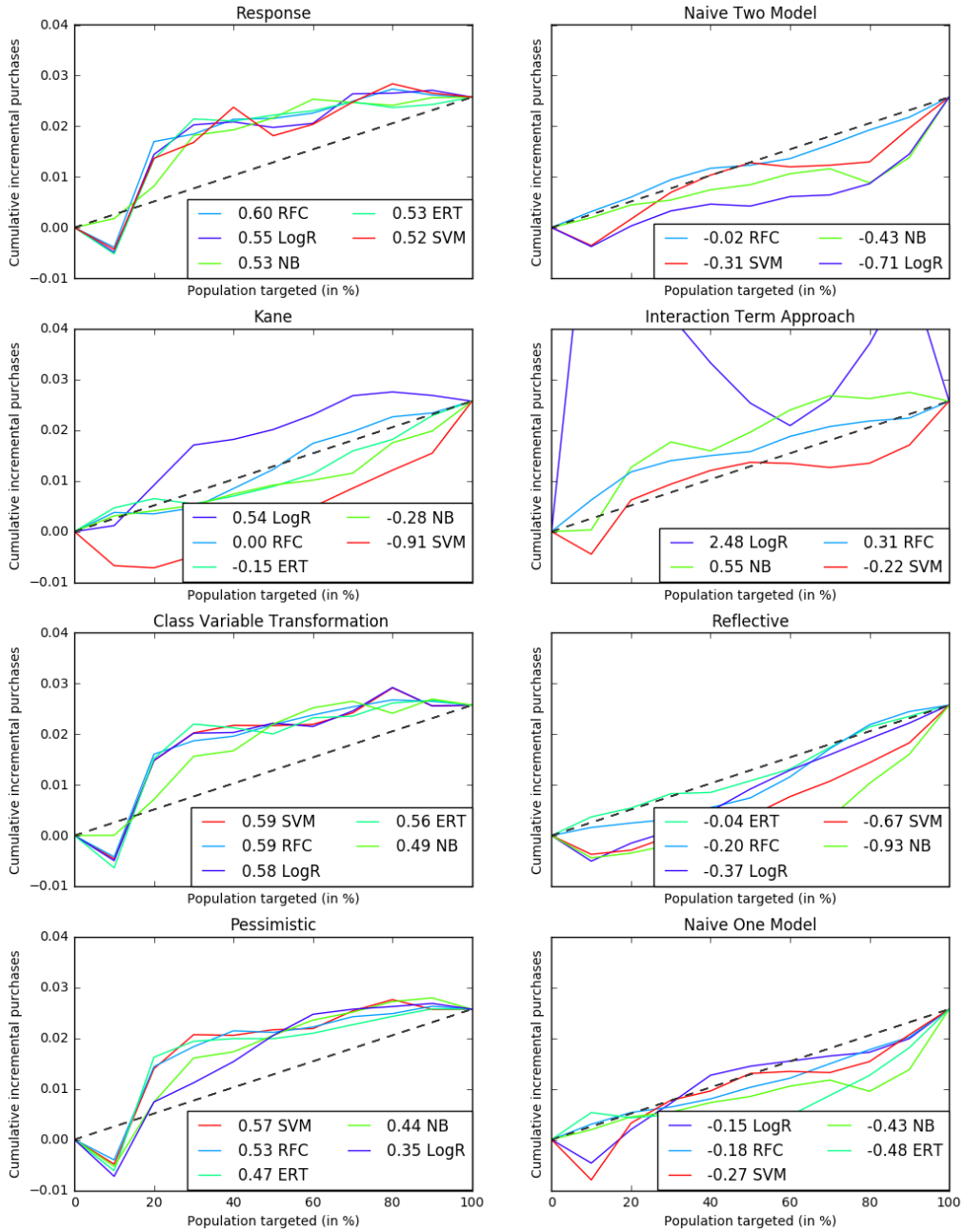

*Figure 19: Qini curves from the **second** validation set of the **first** reduced dataset.*

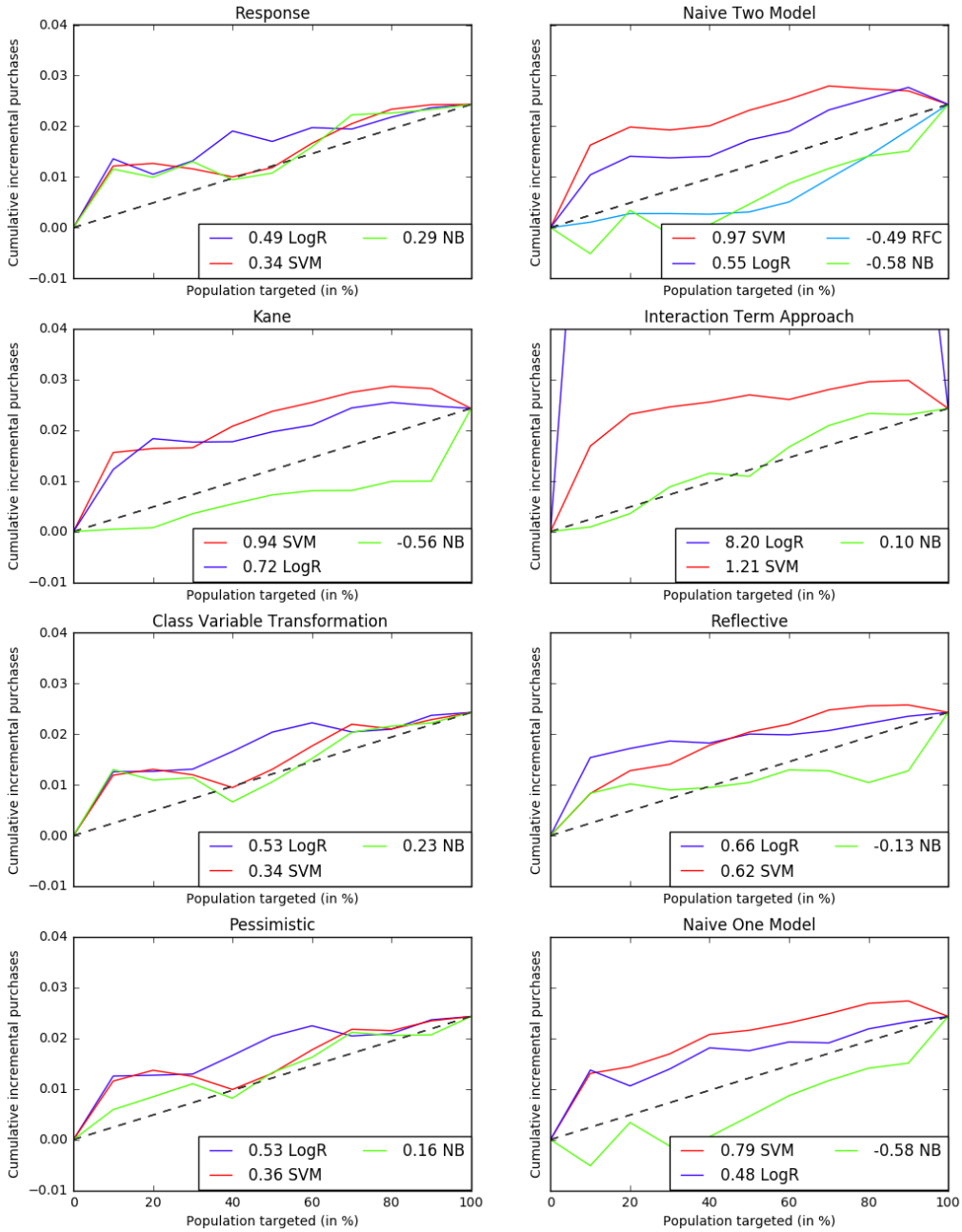

*Figure 20: Qini curves from the **first** validation set of the **second** reduced dataset.*

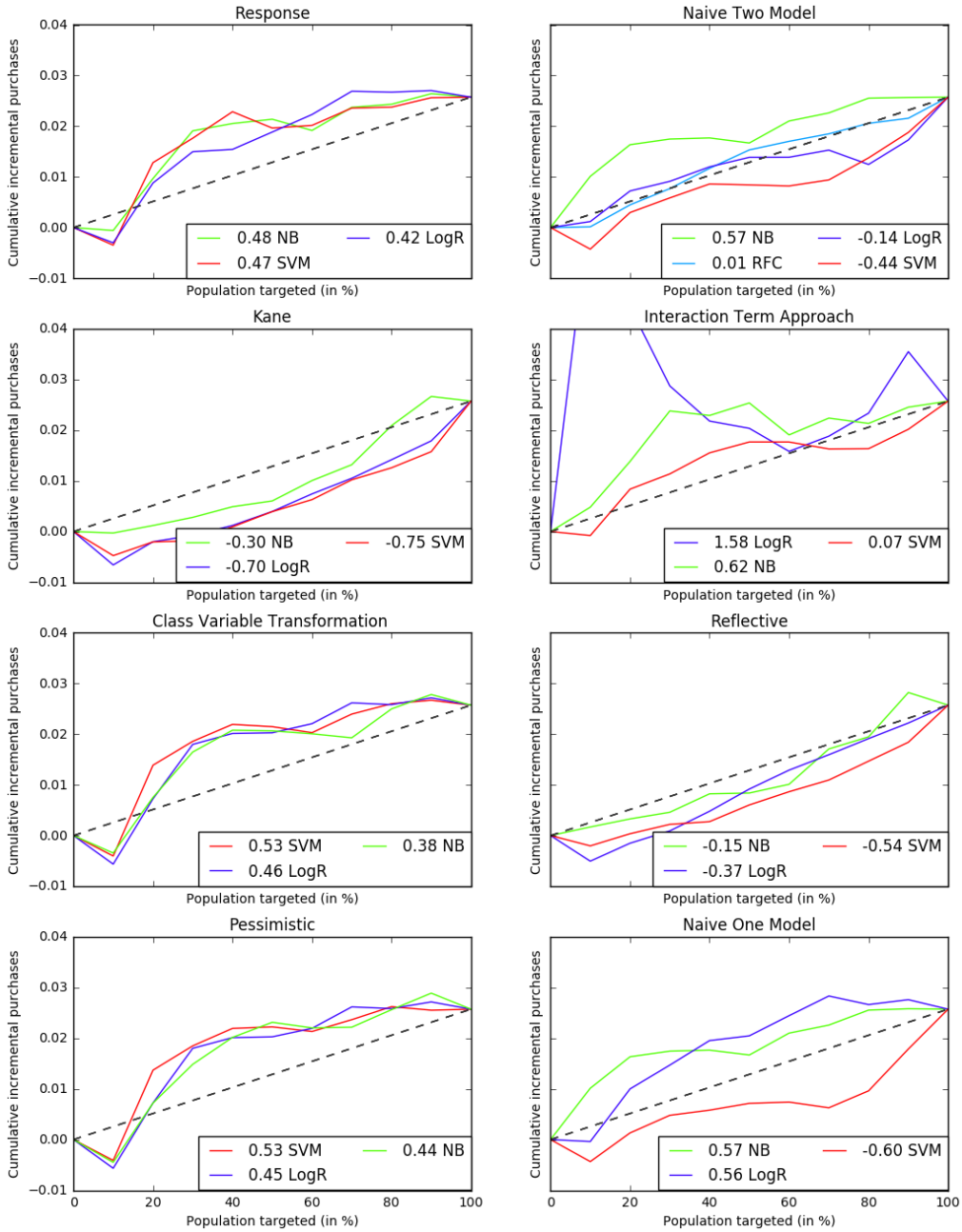

*Figure 21: Qini curves from the **second** validation set of the **second** reduced dataset.*