Profit-based classification in customer churn prediction

Ashkan Zakaryazad

Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, USA
ashkan.zakaryazad@gatech.edu

Taewoon Kong

Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, USA
twkong@gatech.edu

1 Problem Statement

Churn prediction is one of the common applications of the classification in the business settings. The word "churn" means to stop consuming products of a specific company and use fungible product of another company because of its better quality or service or less price. There are lots of studies such as (A. D. Athanassopoulos., 2000; C. B. Bhattacharya, 1998; M. Colgate. et al., 1996) which show that acquiring a new customer for a company is five or six times more expensive than retaining an existing one. Accordingly, nowadays most of the financial institutions are concerned with customer retention studies to prevent losing their market share and maximize their gained profit from existing customers. The primary objective of customer retention is to maximize the potential profit which can come from existing customers. In most of the churn prediction studies, the objective of classification is to minimize the prediction error and accordingly maximize the accuracy of the prediction. This approach is definitely an optimal approach when the objective is to correctly classify the customers as much as possible, however, it may reach suboptimal solution when the objective is to maximize the profit of churn prediction for the company. In our case, the bank has information about customers' lifetime value for the next period (one year) which can be used as a profit metric to show the importance of each of the customers. In this study, we have two objective:

1. Developing a profit-based classification algorithm which classifies churners and non-churners such that it maximizes the total potential profit of the bank by giving more weight to detection of profitable churner customers.

2. Finding appropriate individual incentive offer value for each of the churner customers instead of giving fixed offers to all of them to ensure that more profitable customers are getting more valuable offers than other churners and accordingly minimize their corresponding churn (leaving) probability.

2 Data Source and Description

In this study, we gathered the data set from a well-known Turkish bank. There are totally 20000 samples (customer), each of which has 24 attributes (features) where one of them is response (dependent variable) and 23 are predictors (independent variables). Recency, frequency, and monetary value of a customer have proven to be powerful predictors (Rossi, 1996) however, our data set includes one demographic information (age) as well and totally we have four types of predictors. Since the training prediction error is not a good estimator of test error, we separated the data set to two subsets. To do so, we randomly choose 70% of data set as training and the rest for testing the accuracy of each of the algorithms. following table describes each of the variables of bank data set which are used in this study.

Table 1: Overview of variables in the analysis

#	*Name of the variable*	*Description*
1	CLV	*Customer Lifetime Value*
2	max_bal_check	*Max balance of checking account*
3	num_inct_mo	*# of inactive months*
4	avg_trcn_6	*Average of cheking transaction for last 6 months*
5	num_check_acnt	*# of checking accounts*
6	avg_check_3	*Average of checking balance for last 3 months*
7	max_w_mo_ago	*Max wealth (checking+saving) corresponds to how many months ago?*
8	max_lbl_mo_ago	*Max liability corresponds to how many months ago?*
9	dff-lbl-from_pst_mo	*Difference of liablity from past month ($)*
10	dff-w-from_pst_mo	*Difference of wealth from past month ($)*
11	max_w	*Max wealth ($)*
12	max_lbl	*Max liabliity ($)*
13	avg_num_act_acnt_6	*Average # of active accounts-last 6 months*
14	avg_act_acnt	*Total average # of active accounts*
15	avg_over_drft	*Average of overdraft amount ($)*
16	avg_over_drft_6	*Average of overdraft amount -last 6 month ($)*
17	avg_cc_3	*credit card used amount-average of last 3 month*
18	tot_cc	*Total credit card used amount*
19	save_bal_end	*Balance of saving acount at the end of month ($)*
20	diff_save_past_mo	*Difference of saving account from past month*
21	diff_sec_past_mo	*Difference of securities from past month ($)*
22	avg_sec_last_mo	*Average of type securities balance within last month ($)*

CLV is one of the most important variables in this study which will be used to compute the total profit of the classification. We used Linear regression model to predict it for the test examples however the i.i.d normality assumption of the error is violated which is the case in most of the finance data set. Then we decided to use Quantile regression which is appropriate for the cases where the variance of the prediction error is correlated with the input variables.

2.1 Quantile regression (QR) for CLV prediction

Linear regression model focuses on the conditional mean function which means this model describes the variation of mean of response (y) with the vector of predictors. Since, linear regression works based on Least-Square (LS) approach, it has some basic assumptions about the model. In LS, we assume that the prediction error is normally distributed with constant variance. By taking these assumption we state that disregarding the value of x. So the predictors are assumed to affect only on the location of conditional distribution $E(y|x)$ not its dispersion. However, in most of the financial data, this is not the case when we are predicting an amount (profit/cost) and the variance of the dependent variable increases with the mean which violated the assumption of homoscedasticity. Quantile regression uses conditional quantiles of the response variable and let us to do further analysis on relationship of predictors and different quantiles of distribution of the response variable. Figure (1) shows an example of distribution of response variable against X_{10}(maximum wealth).Since the variance of CLV is positively correlated withX_{10}, slope coefficients for each of the $\tau - th$ quantiles ($\tau \in (0.1, 0.9)$) are different (blue lines). The generated line by OLS method is also represented by red color.

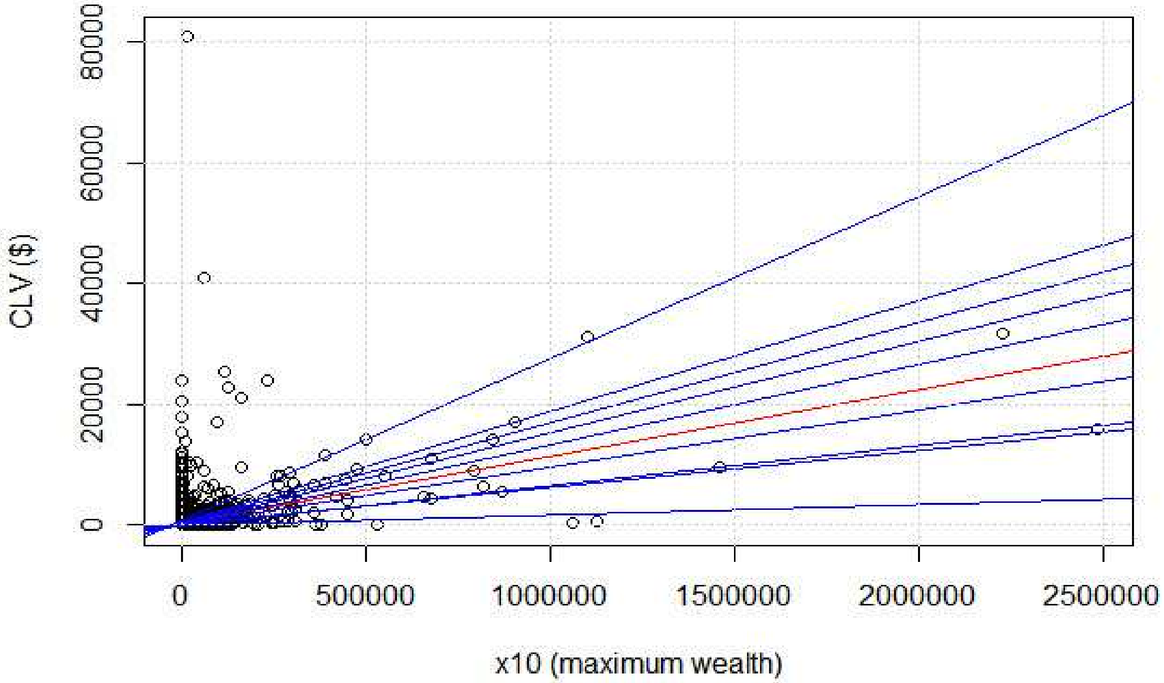


Figure 1: Illustration of quantile regression for 25-th, 50-th, and 75-th quantile.

Any real-valued random variable, X, can be expressed by its cumulated distribution (right-continuous) function (CDF) as $F(x) = P(X \leq x)$

Then for each any τ, $0 < \tau < 1$

$$F^{-1}(\tau) = inf\{x \,:\, F(x) \geq \tau\} \tag{1}$$

is called $\tau - th$ quantile of X. For instance the median of X is $F^{-1}(0.5)$ which is the center of distribution where half of the points are on its left and half of points are on its right. The problem of finding $\tau - th$ quantile $\xi(\tau)$ can be written as :

$$\xi(\tau) = \underset{\xi \in R}{argmin}(i = 1)n\sum \rho_\tau(y_i - \xi) \tag{2}$$

where $\rho_\tau(z) = z(\tau - I(z > 0))$ and $I(.)$denotes the indicator function. the loss function ρ_τ assigns a weight of τ to positive residuals and a weight of $(1 - \tau)$ to negative residuals.

Ordinary Least Square (OLS) method in linear regression finds the conditional mean of response variable (y) by solving:

$$\underset{\mu \in R}{min}(i = 1)n\sum (y_i - \mu)^2 \tag{3}$$

which suggests that if we would express the conditional mean of y given x as $\mu(x) = x'\beta$ then we can estimate β by solving:

$$\underset{\beta \in R}{min}(i = 1)n\sum (y_i - x_i'\beta)^2 \tag{4}$$

In median regression, we proceed in exactly the same way, but here, we try to find an estimate of β that minimizes the sum of the absolute deviations:

$$\underset{\beta \in R}{min}(i = 1)n\sum |y_i - x_i'\beta| \tag{5}$$

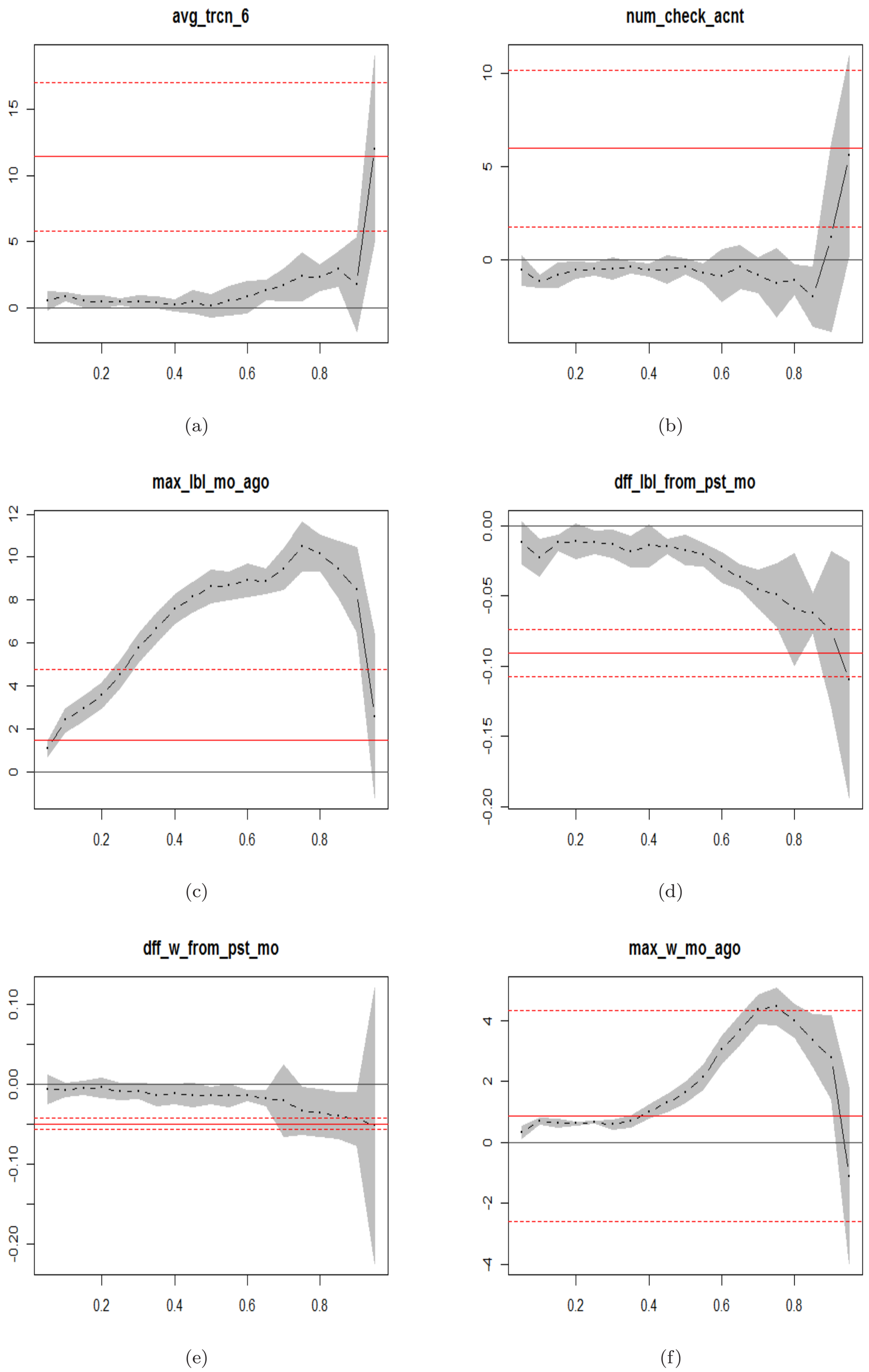
If we extend the median case to all other quantiles of interest, the result will be quantile regression. By using loss function mentioned in Equation (3), the linear conditional quantile function extends the $\tau - th$ sample quantile $\xi(\tau)$ to regression setting in the same way as the linear conditional mean or median function:

$$\beta_{hat}(\tau) = \underset{\beta \in R}{argmin}(i = 1)n\sum \rho_\tau(y_i - x_i'\beta) \tag{6}$$

for any quantile $\tau \in (0, 1)$. $\beta_{hat}(\tau)$ is called the $\tau - th$ regression quantile and in the case of $\tau = 0.5$, it minimizes the sum of absolute residuals which corresponds to median regression. For further detail explanation of the quantile regression Koenker's book (2005) is an appropriate resource.

2.1.1 Customer lifetime value

Although much research on customer lifetime value has employed conventional least-squares regression methods, it has been recognized that the resulting estimates of various effects on the conditional mean of CLV were not necessarily indicative of the size and nature of these effects on the lower or upper tail of the CLV distribution. A more complete picture of covariate effects can be provided by estimating a family of conditional quantile functions.



(a)

(b)

(c)

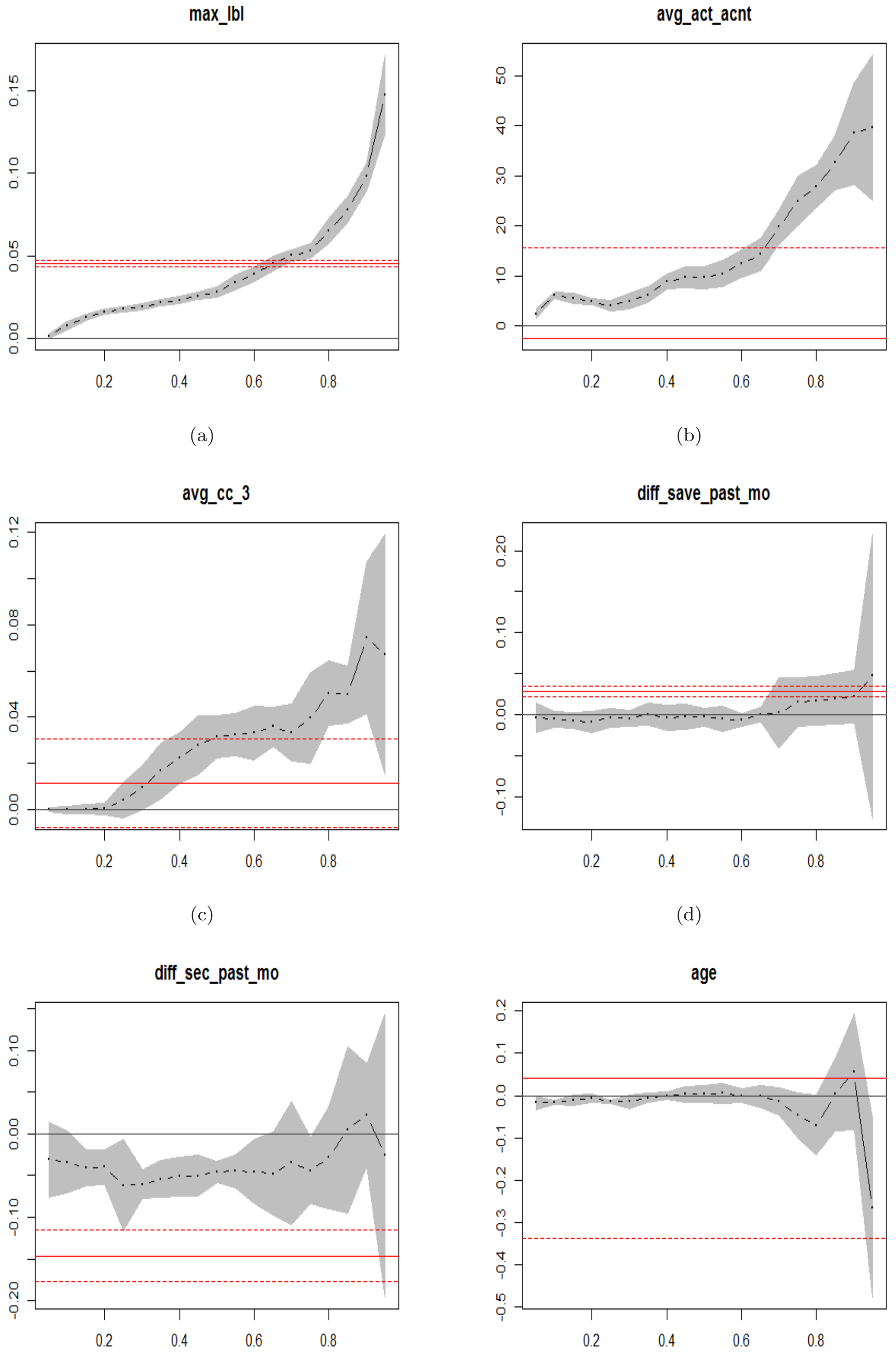
(d)

(e)

(f)

Figure 2: Quantile regression plots for CLV prediction. Effects or Beta's (vertical axis) against quantiles of CLV (horizontal axis).

Following figure shows some of the predictors in the quantile regression analysis which have different effect on CLV prediction than least square model. Each plot depicts one variable in the quantile regression model, $\{\beta_{hat}(\tau) : j = 1, \dots, 22\}$. Note that the plots in Figure 5 are



(a) (b)

(c) (d)

(e) (f)

Figure 3: Quantile regression plots for CLV prediction. Effects or Beta's (vertical axis) against quantiles of CLV (horizontal axis).

obtained using Bayesian estimation with vague priors on the unknown model parameters. The plotted point estimates and the credible intervals are the expectation, $Q_{.025}$ percentile and $Q_{.975}$ percentile obtained from the marginal posterior distribution of the different parameters. The solid line with filled dots represents the point estimates of the regression coefficients for the different quantiles, $\tau_q : q = 0.05, \ldots, 0.95$, by 0.5. The lightly shaded area depicts a 95% confidence interval for estimated effects (β's). Superimposed on the plot is a dashed line representing the ordinary least-squares estimate of the mean effect, with two dotted lines representing a 95% confidence interval.

A first glance at the above plots suggests that for predicting the CLV it might be worthwhile to split up the effects of predictors in three distinct parts $\tau \in [0, 0.3]$, $\tau \in [0.3, 0.8]$, $\tau \in [0.8, 0.9]$. Because 0.3 and 0.8 quantiles are start or end points of differences between linear and quantile model. This conclusion is proved by the results of comparing these different models in appendix B1. The three models for corresponding three quantiles are significantly different with $p - value < 2.2 \times 10^{-13}$. In the two first plot we can see that OLS overestimates the effect of *avg_trcn_6(Average transactions in the last six months)* and *num_check_acnt(number of checking accounts)* on the CLV for quantiles less than 0.9, however, these two variables are not even significant predictors for the first and middle quantiles. For the third predictor (*max_lbl_mo_ago*) OLS finds it insignificant which is significant with high effects for middle and upper quantiles (from 0.3 to 0.9). For two predictors *"difference of liability from the past month" (dff-lbl-from_pst_mo)* and *"difference of wealth from past month" (dff-w-from_pst_mo)*, OLS gives more importance to it for quantiles less than 0.8 but after 0.8 quantile both models reach same result. "maximum wealth" (amx_w) has actually more effect on predicting the CLV of profitable customers (greater than 0.5 quantile) but not even significant for less profitable customers (less than 0.2 quantile). The predictor "Maximum liability" (max-lbl) is quite same as max_w and its effect on CLV prediction increases with profitability of the customers. We can interpret all of these plots in the same manner but the surprising result is that the predictor "Age" has been introduced as an important variable in predicting the CLV of the customers, however, in bank data set it is not significant in predicting the CLV of the customers.

The result of the quantile regression analysis is that we can use three different regression models for three different types of customers. One for less profitables, one for middle quantiles and one for most profitable customers. However, managers would target the most profitable customers in their database so they have to use the model for higher quantiles of the CLV with its corresponding effects. We used higher quantile model for predicting the CLV of the customers because we are more interested in profitable customers.

3 Classification by Ensemble Method

The main objective of this study is to maximize the total profit instead of minimizing classification error. For the purpose we build a modified Ensemble algorithm focusing on maximizing profits instead of minimizing classification errors and it is one of the main contributions of this study.

Ensemble methods use multiple learning algorithms to obtain better performance than one could be obtained from any of the constituent learning algorithms (Opitz, D. et al., 1999; Polikar, R., 2006; Rokach, L., 2010). Among many methods, Boosting is one of the most commonly used ensemble method (L, Breiman., 1996; Z. Zhi-Hua., 2012). It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. While boosting is not algorithmically constrained, most of the Boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learners' accuracy. After a weak learner is added, the data is re-weighted: examples that are misclassified gain weight and examples that are classified correctly lose weight. Thus, future weak learners focus more on the examples that previous weak learners misclassified, which makes the boosting method involve incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. Note that there are two kinds of weights in Boosting. The one is for each weak classifier, and the another one is for each sample.

Recall that we need to modify the established Boosting algorithm to maximize profits instead of minimizing classification errors. Even if it is the best to develop a totally new weighting way in terms of profits, it is too complex to achieve in the time we have been given. Thus, we achieve our goal to only modify the way of initial weighting. In fact, the original method initially gives equal weights to each sample. However, we give the initial weights to each sample based on its corresponding CLV. If one sample has a large CLV, it is given a large weight and vice versa. Details are included in section 3.1. We are able to approximately maximize profits by this simply modified initial weighting method. As a final comment, in this study, although there are many boosting algorithms, we use AdaBoost which is so adaptive that take full advantage of the weak learners.

3.1 AdaBoost

```
**Data:** $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{0, 1\}$
**Result:** The final hypothesis: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
Initialize $D_1(i) = \frac{CLV_i}{\sum_{i=1}^{m} CLV_i}$,    $i = 1, \ldots, m$;
**for** $t = 1, \ldots, T$ **do**
    1. Train weak learner using distribution $D_t$;
    2. Get weak hypothesis $h_t : X \to \{0, 1\}$
       with error $\epsilon_t = Pr_{i \sim D_i}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$;
    3. Update: $D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ where $Z_t$ is a normalization factor (chosen so
     that $D_{t+1}$ will be a distribution);
**end**
```

Algorithm 1: Pseudocode for AdaBoost

The AdaBoost algorithm, short for 'Adaptive Boosting', is a machine learning meta-algorithm and introduced in 1995 by (Y. Freund. et al., 1995), solved many of the practical difficulties of the earlier boosting algorithms. Pseudocode for AdaBoost is given in Algorithm 1. The algorithm takes as input a training set $(x_1, y_1), \ldots, (x_m, y_m)$ where each x_i belongs to some domain or instance space X, each label y_i is in some label set Y, and the number of samples in training set is m. For most of this paper, we assume $Y = \{0, 1\}$ since the response variable of our data is binary. AdaBoost calls a given weak or base learning algorithm repeatedly in a series of rounds $t = 1, \ldots, T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted $D_t(i)$. Originally, all initial weights, $D_1(i)$, are set equally as $\frac{1}{m}$ as in Algorithm 1, however, we set the initial weights based on CLV values of each sample as follows:

$$D_1(i) = \frac{CLV_i}{\sum\limits_{i=1}^{m} CLV_i}, \qquad i = 1, \ldots, m. \tag{7}$$

After a first iteration based on the initial weights, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set.

The weak learners job is to find a weak hypothesis $h_t : X \to \{0, 1\}$ appropriate for the distribution D_t. The goodness of a weak hypothesis is measured by its error

$$\epsilon_t = Pr_{i \sim D_i}[h_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} D_t(i). \tag{8}$$

Notice that the error is measured with respect to the distribution D_t on which the weak learner was trained. In practice, the weak learner may be an algorithm that can use the weights D_t on the training examples. Alternatively, when this is not possible, a subset of the training examples can be sampled according to D_t, and these (unweighted) resampled examples can be used to train the weak learner.

Once the weak hypothesis h_t has been received, AdaBoost chooses a parameter α_t as in Algorithm 1. Intuitively, α_t measures the importance that is assigned to each h_t. Note that $\alpha_t \geq 0$ if $\epsilon_t \leq \frac{1}{2}$ (which we can assume without loss of generality), and that α_t gets larger as ϵ_t gets smaller.

The distribution D_t is next updated using the rule shown in Algorithm 1. The effect of this rule is to increase the weight of examples misclassified by h_t, and to decrease the weight of correctly classified examples. Thus, the weight tends to concentrate on hard-to-classify examples. The final hypothesis H is a weighted majority vote of the T weak hypotheses where α_t is the weight assigned to h_t. R.E. Schapire. et al. (1998) show how AdaBoost and its analysis can be extended to handle weak hypotheses which output real-valued or confidence-rated predictions. That is, for each instance x, the weak hypothesis h_t outputs a prediction $h_t(x) \in \mathbb{R}$ whose sign is the predicted label $(-1, +1)$ and whose magnitude $|h_t(x)|$ gives a measure of confidence in the prediction. In this paper, however, we focus only on the case of binary, $Y = \{0, 1\}$, valued weak-hypothesis predictions.

3.1.1 Analyzing the training error

The most basic theoretical property of AdaBoost concerns its ability to reduce the training error. Let us write the error ϵ_t of h_t as $\frac{1}{2} - \gamma_t$. Since a hypothesis that guesses each instances class at random has an error rate of $\frac{1}{2}$ (on binary problems), γ_t thus measures how much better than random are h_ts predictions. R.E. Schapire. et al. (1997) proved that the training error (the fraction of mistakes on the training set) of the final hypothesis H is at most

$$\prod_t [2\sqrt{\epsilon_t(1-\epsilon_t)}] = \prod_t \sqrt{1-4\gamma_t^2} \leq exp(-2\sum_t \gamma_t^2). \tag{9}$$

Thus, if each weak hypothesis is slightly better than random so that $\gamma_t \geq \gamma$ for some $\gamma > 0$, then the training error drops exponentially fast.

A similar property is enjoyed by previous boosting algorithms. However, previous algorithms required that such a lower bound be known a priori before boosting begins. In practice, knowledge of such a bound is very difficult to obtain. AdaBoost, on the other hand, is adaptive in that it adapts to the error rates of the individual weak hypotheses. This is the basis of ‘Ada’ is short for ‘adaptive’. The bound given in Equation 9, combined with the bounds on generalization error in R.E. Schapire. et al. (1998), prove that AdaBoost is indeed a boosting algorithm in the sense that it can efficiently convert a weak learning algorithm (which can always generate a hypothesis with a weak edge for any distribution) into a strong learning algorithm (which can generate a hypothesis with an arbitrarily low error rate, given sufficient data).

3.2 Simulations

For the weak learners, we consider Logistic Regression, Artificial Neural Network (ANN), Support Vector Machines (SVM), Decision Tree, Naive Bayes, Discriminant Analysis (LDA or QDA), and k-NN. Also, we add a random classifier to show a clear decreasing pattern after iterations.

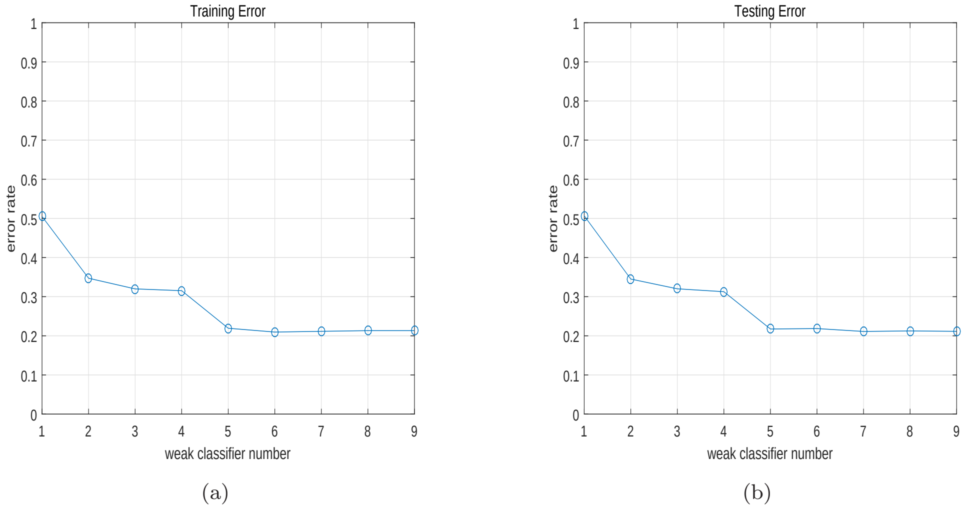
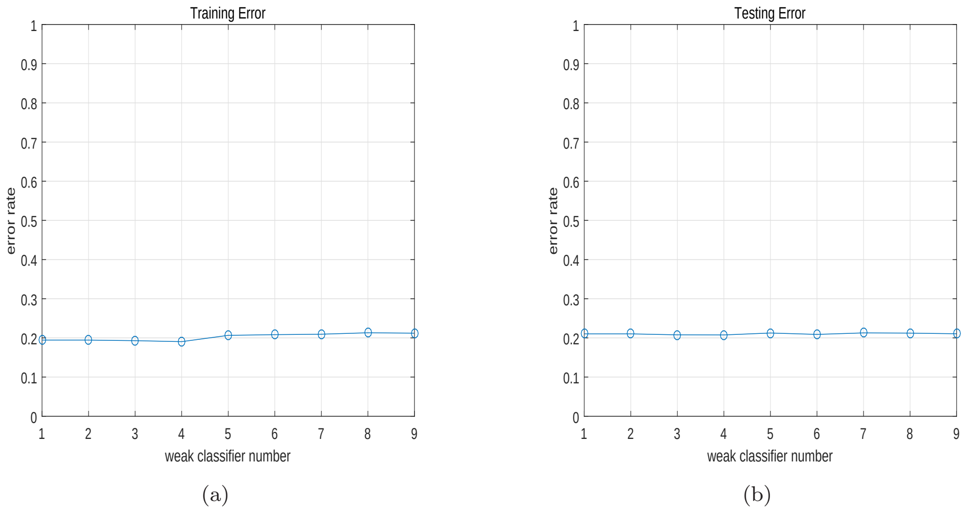


Figure 4: (a) Training errors based on AdaBoost when the random classifiers are added first. (b) Test errors based on AdaBoost when the random classifiers are added first.



(a)

(b)

Figure 5: (a) Training errors based on AdaBoost when the classifiers which are very well-adaptive to current data are added first. (b) Test errors based on AdaBoost when the classifiers which are very well-adaptive to current data are added first.

See the Figure 4(a) and 4(b). The first one illustrates training errors and the latter testing errors. Adding weak classifiers, the both errors clearly decrease to a certain extent. The reason why it shows the decreasing pattern is that the worst weak learner 'Random classifier' is added first. Even if it shows a clearly decreasing pattern, the result with adding the worse classifiers first is not inclined to be the best. Thus, if possible, we avoid to adding the worse classifiers first.

Conversely, if other good weak learners are added first, it is likely to show a non-changing pattern as in Figure 5(a) and 5(b). This is because it is hard for the weak classifiers added after the classifiers which are very well-adaptive to current data to provide clearly useful information. Even if it does not show a decreasing pattern, the result with this case is often inclined to be the best. However, it is not always guaranteed to occur this kind of lucky situation. Thus, utilizing the AdaBoost is still useful since it can guarantee the best result in almost all situations.

We discovered the best combination of weak classifiers through simulations. Also, we derive that QDA and Naive Bayes are relatively worse than other 6 methods. Thus, adding them at the end of iterations would provide a better result. As discussed in section 3.1, the weighted strong classifier is obtained after applying Adaboost method and the final profit-based AdaBoost model we discovered is as follows:

$$\begin{aligned} H(x) &= 0.1403 * LogisticRegression(x) + 0.1400 * SVM(x) + 0.1469 * ANN(x) \\ &\quad +0.1377 * LDA(x) + 0.0704 * QDA(x) + 0.0827 * NaiveBayes(x) \\ &\quad +0.1531 * DecisionTree(x) + 0.1425 * kNN(x) \end{aligned}$$

Note that the weights for QDA and Naive Bayes are relatively small and those for others are equally large.

Next, we calculate test errors and total CLV of correctly classified customers based on the 8

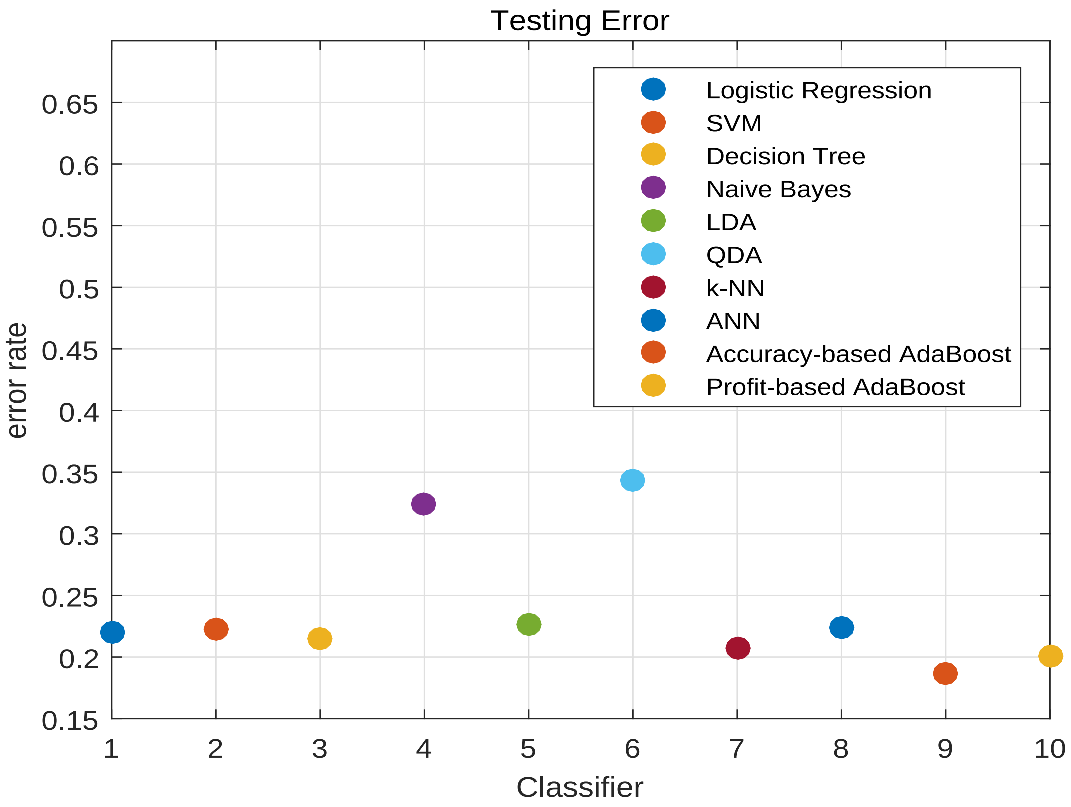


Figure 6: Testing errors by all 10 methods are illustrated. Note that the accuracy-based AdaBoost shows the lowest error.

weak learners, the accuracy-based AdaBoost classifier, and the profit-based AdaBoost classifier in Figure 6 and 7. First, we see that the testing error by the accuracy-based AdaBoost is the lowest in Figure 6. It proves that the accuracy-based AdaBoost outperforms other classifiers in terms of error. Similarly, we see that total CLV of correctly classified customers by the profit-based AdaBoost is the largest in Figure 6. Thus, it also proves that the profit-based AdaBoost outperforms other classifiers in terms of total CLV of correctly classified customers. This is quite interesting because we can easily obtain these results with only changing the initial weights for samples. Thus, we can conclude that we achieve our objective of approximately maximizing profits. The way to achieve totally maximizing profits will be discussed in section 5

4 Profit Maximization by Incentive Offer Selection

In the post-processing step of the study we maximize the total profit of customer retention with solving an optimization problem to find individual incentive offer for each of the customers and make sure that the offer selection system works fairly considering customers' profit for the company. We will compare two policies about retention promotions (offers). First of them is to giving some fixed offers to all of the target customers (predicted as churner) and second, the variable incentive offer for each of the customers. The objective is to maximizing the profit of the company by minimizing the churn probability of customers. Different offers may effect differently on customers churn probability.

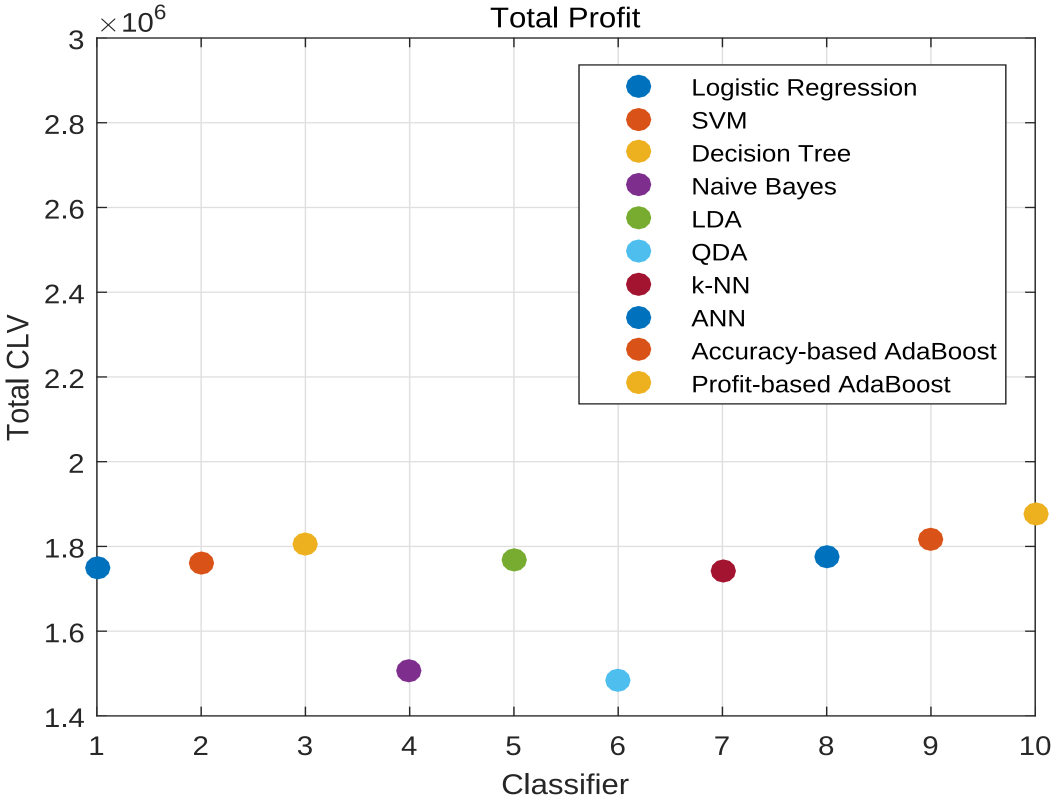


Figure 7: Total CLV of correctly classified customers by all 10 methods are illustrated. Note that the profit-based AdaBoost shows the largest total CLV.

4.1 Expected Profit of Churn Prediction

In the Profit-based classification approach, the first important point to be considered is to have a base scenario which represents the system without using the recommended models. In the customer churn prediction, the base scenario is that there is no prediction system and churner customers are not detected by the company and leave. Consequently, the company loses all of the potential profit which could be earned from those customers. Accordingly, all of the profits and costs in this study have been calculated based on base scenario which assumes that there is no churn prediction system. For instance, for the correctly detected churner customers, we consider a profit which is a proportion of his/her life time value and also a cost which is related to the retention promotion made for that customer. These profits and costs are not counted in the base scenario. Take another instance and consider false negative instance in the system, those who are actual churners but predicted non-churners in the prediction system. Although we lose that customer's potential profit, we do not consider it as a cost because in the base scenario this customer's profit was lost as well and we have no added cost comparing with the base scenario.

According to this base scenario and individual profits and costs, we can assume a net profit matrix for each instance. Assume that instance is an actual churner (positive) instance and is an actual non-churner (negative) one. The individual net profit matrix is as following:

CLV_i is the customer life time value for the instance (customer) i, $S_{i,old}$ is the score which

Table 2: Individual net profit matrix for churn prediction

		Actual	
		Instance i (churner)	Instance j (non-churner)
Predicted	Churner	$CLV_i(S_{i,old} - S_{i,new}) - c_i$	$-c_j$
	Non-churner	0	0

comes from the model and it is assumed to be the predicted churn rate of the customer i, $S_{i,new}$ is the churn probability of the customer i after making an promotion offer to him/her and it is calculated based on equation 10. c is the fixed cost of offer which will be paid for all of predicted (true and false) positive instances. The profit will be the change in the churn rate multiplied by the CLV of the customer. We assume that in all of the cases. In other words, after the retention promotion is given, the customer's churn rate will be decreased or remain the same.

Calculating the amount of net profit for churn prediction needs a post-processing analysis and we have to consider the effects of each incentive offers on the predicted churner customers. For this purpose, we benefit from domain experts' opinion to quantify the effects of each incentive offer for different values of churn probability. For each of the possible churn probabilities, the expert gives the new changed customer churn probability assuming a specific kind of incentive offer. For different values of churn probabilities and also variable incentive offers, we made an approximation with fitting tools and found the relationship between them. The result revealed that the relationship is like a sigmoid function with different parameters for each of the churn probabilities. The new churn probability is calculated as:

$$S_{new} = \frac{2S_{old}}{1 + S_{old}^{-x}} \tag{10}$$

In this equation S and x represent the churn probability of each customer and cost parameter of incentive offer.We consider different incentive offers as input of sigmoid function and the range as its output which shows the previous (initial) and new churn probability of the customer after receiving an incentive offer. The relationship for some examples with different churn probabilities are depicted in the figure 8:

The above figure is of interest in itself, because it represents the behavior of different types of customers in terms of churn probability regarding variable incentive offers. The reaction of customers with high churn probability shows that their churn probability is decreasing very slightly and slowly comparing with other ones and small offers cannot make significant change in their decision to leave or stay at company. On the other hand, more loyal customers who has lower churn probabilities has better reactions regarding even small incentive offers. This relationship helps us to appropriately find the net profit of churn prediction model regarding different types of incentive offers.

Moreover, to boost our approximation about customers' profitability, we made another approximation using different life time values of customers to show the behavior of profit function regarding the variable incentive offers. In this approach we not only use the churn probability of the customer as our input variable, but also consider his/her particular profit (life time value) to

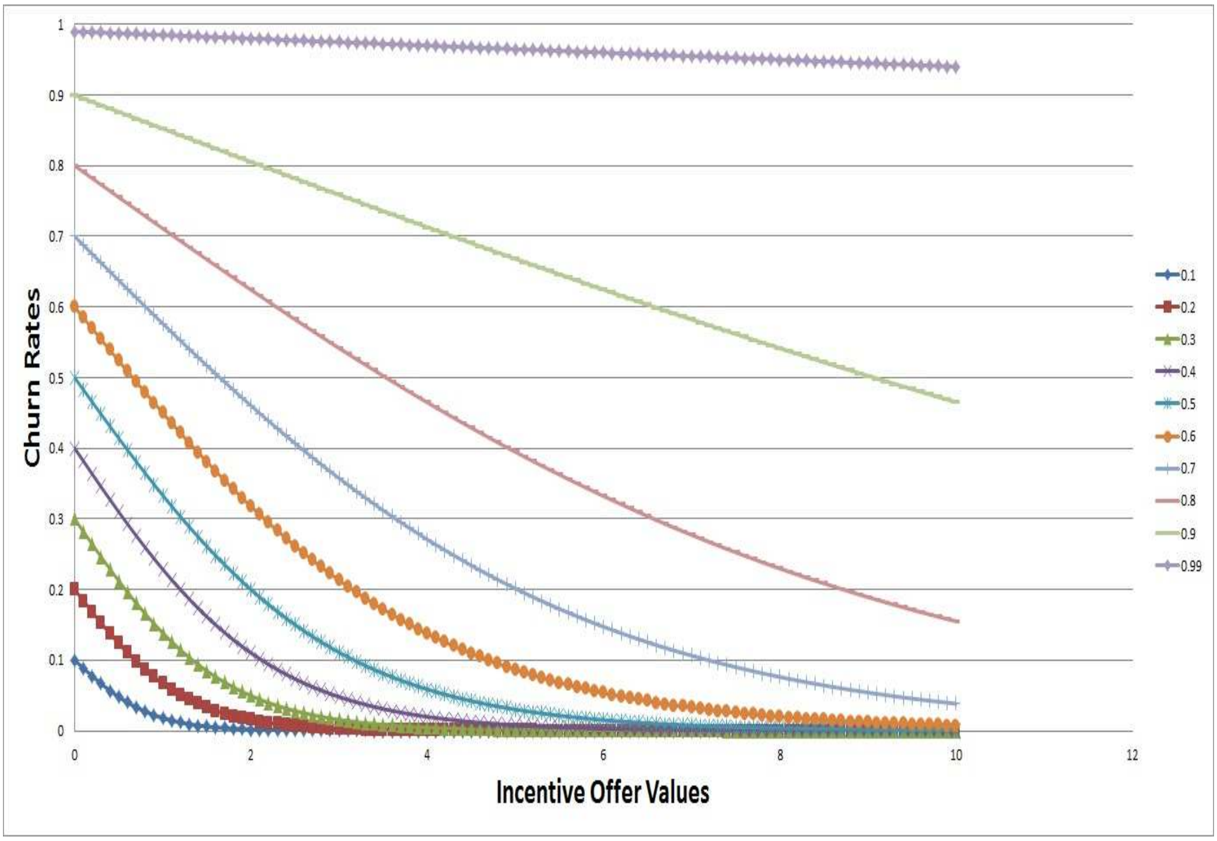


Figure 8: Sigmoid relationship between incentive offer cost and customer churn rate.

find the total profit of each customer when selecting a specific kind of incentive offer to make for him/her. The result is a relationship between incentive offers and customer's churn probability, but the difference here is that the relationship shows the amount of money earned for customers with particular churn probability when selecting different incentive offers. The relationship is depicted in the following figure for two example customers from data set with different churn probabilities and life time values. The first instance is a customer with churn probability 0.9 and life time value of 35000. The second one is a customer with churn probability 0.6 and life time value of 10000.

Figure 9 shows that, there is possibility to a less loyal customer to have more profit for company than a more loyal one for a specific incentive offer. For example in this comparison, if company gives offers which costs more than 2.1 unit money, the customer with higher churn probability will have more profit than the other one. This result confirms the results for Reinartz and Kumar (2000) which shows that loyal customers are not necessarily the most profitable customers to the company. Selecting an incentive offer depends on company's budget which has been assigned to customer relationship management projects and a budget constraint has to be considered for this purpose. The incentive offer policy is an issue in which the managers of the company have to make decision for. For example some companies prefer to give offers with fixed amount of money for all of the customers (fixed-incentive). In this scenario, it is necessary to find a point in the X-axes (incentive offer cost) in which the total profit is maximized. Some others prefer to give variable-cost incentive offers for each of the customers considering their profit and

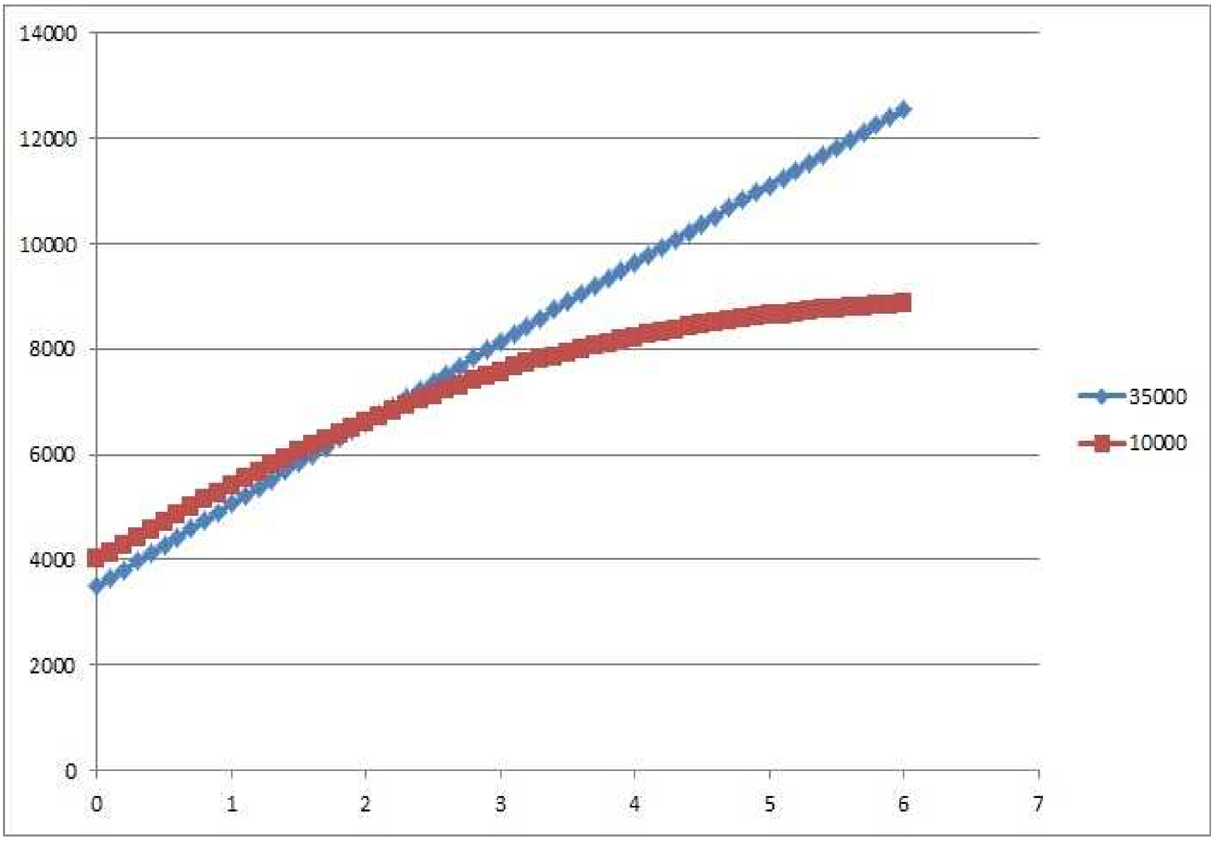


Figure 9: Relationship between incentive offer cost and customers retention profit.

churn probability (variable-incentive). In the latter scenario, we have to find a maximum point in total profit for each of the customers and use the corresponding incentive offer for each of them.

4.2 Fixed Retention Promotion

If the financial institution decides to give same retention promotion to all of the targeted customers, the total profit for customer retention has to be calculated and all kinds of promotion costs have to be considered and the promotion which maximizes the total profit has to be selected for all of the customers. The total profit here means the sum of all individual profits of each of the customers. Also, budget constraint has to be considered which means the total cost of retention promotions has not to exceed the budget assigned for this project. The formulation of this scenario is as following:

$$MAX(P) = \sum_{i=1}^{n}(S_i - \frac{2S_i}{1+(S_i)^{-y}})CLV_i - \sum_{i=1}^{n} c_y \tag{11}$$

$$\sum_{i=1}^{n} c_i \leq B \tag{12}$$

In this scenario we offer three types of incentive offer (cash back) for each of the customers. Management will make the decision to give either \$1, \$10 or \$20 offer as a cash back to all of the target customers (which have been classified as churner using the most accurate classifier). The final choice in this scenario is the offer which maximizes the aforementioned objective function. We have applied AdaBoost algorithm to the test set (6000 customer) and the output of the AdaBoost is assumed to be churn probability for each of the customers. We also find the new churn probability of each churner customer as explained with three different offers and the total net profit will be the change in the churn probability which the offer makes on each customer, multiplied by his/her CLV subtracting the cost of the offer. Te results for three incentive offer is as following: The above table shows that one dollar incentive offer (cash back) for all of the predicted churners will be the most profitable decision for the bank when the manager have decided to give equal offers for all of the target customers. Following figure represents the cost and total profit of each of the fixed offers.

Table 3: Profit analysis for fixed incentive offers

Value of the offer($)	Total profit ($)	Number of churners	Total cost of offer ($)	Total Net Profit ($)
1	10902.21	3107	3107	7795.21
10	21568.91	3107	3170	-9501.08
20	31801.21	3107	62140	-30338.79

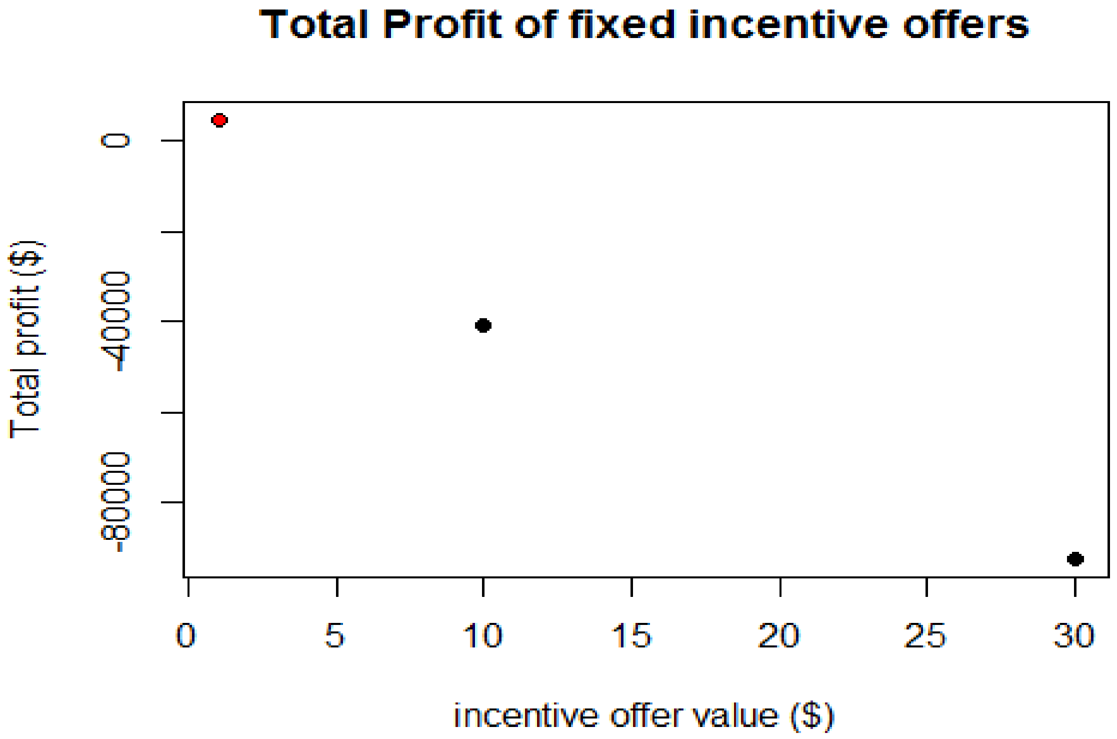


Figure 10: Profit and cost analysis between fixed offers.

4.3 Disproportionate Retention Promotion

In this scenario, there are finite kinds of retention promotions like the previous scenario but here, one customer can get different retention promotion than other one. Therefore, there are some types of promotions available for each of the customers and the type of offer selected for one customer depends on the total profit earned from all of the customers. Here we face a well-known type of optimization problem called integer programing and there are variations of algorithms to solve this problem. The formulation of the problem is as following:

$$MAX(P) = \sum_{j=1}^{J}\sum_{i=1}^{n}(S_i - \frac{2S_i}{1+(S_i)^{-y}})CLV_i - c_j y_{ij} \tag{13}$$

$$\sum_{j=1}^{J}\sum_{i=1}^{n} c_j y_{ij} \leq B \tag{14}$$

$$\sum_{j=1}^{J} y_{ij} \leq 1 \ for \ \forall i \in n \tag{15}$$

$$y_{ij} \in \{0,1\} \tag{16}$$

$$y_{ij} = \begin{cases} 1 & \text{for customer } i \text{ the promotion } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

The aforementioned table represents the results of assigning different promotion offers (cash backs) to different customers. By difference here, we mean that one customer can receive \$1 , \$10, or \$20 promotions based on his/her churn probability and CLV value for the bank. For example following table shows four customers who have receive different types of incentive offers. Following figure represents the cost and total profit of each of the fixed offers compared to the variable (personalized) offer.

Table 4: Profit analysis to compare disproportionate offers with fixed ones

Value of the offer($)	Total profit ($)	Number of churners	Total cost of offer ($)	Total Net Profit ($)
1	10902.21	3107	3107	7795.21
10	21568.91	3107	31070	-9501.09
20	31801.21	3107	62140	-30338.79
variable	29970.90	3107	7319	22651.90

5 Conclusion

In this research we used 8 weak classifier to find a strong classifier by AdaBoost which outperform the formers in terms of both accuracy and total profit. It is proved that the final strong classifier outperforms other weak classifiers in both measures. The one of contributions of this study is that we achieved our goal only with simple change of initial weighting to each sample. In future study, it would be better if we also modify the updating procedure of weights for each sample in terms of maximizing profit. Also, utilizing the AdaBoost in itself is worth considering in any other research fields because almost all main concerns of real life is to find the best method.

Next, we used its output as churn probability of each of the customers to find the actual profit of the churn prediction. The second contribution of this research is the development of

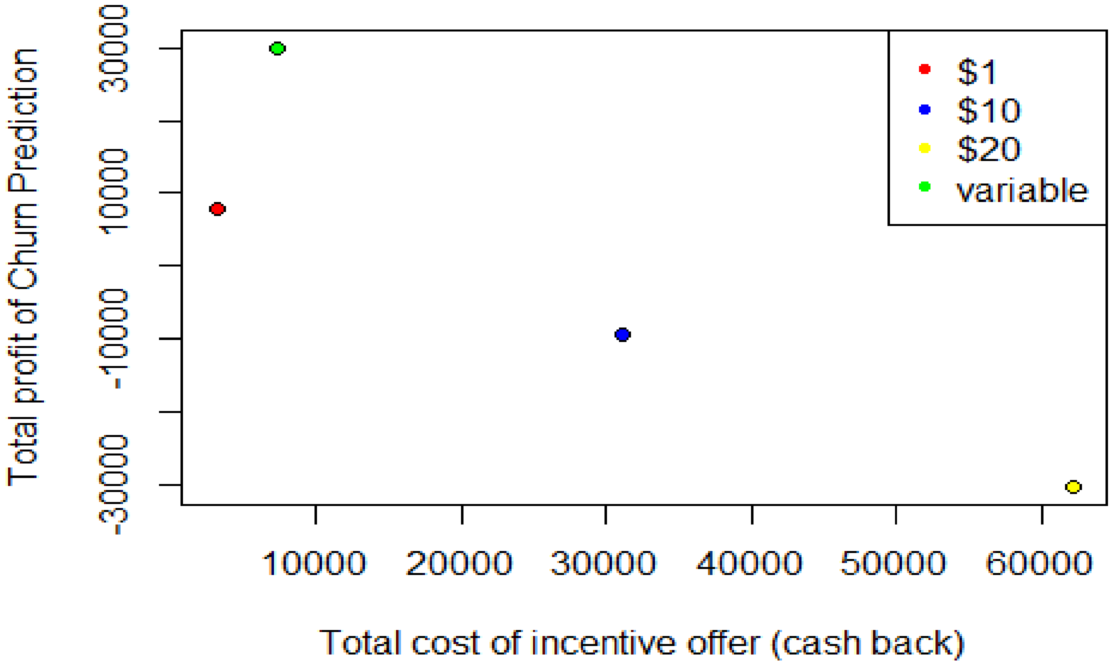


Figure 11: rofit and cost analysis for comparing fixed offers and variable offer.

an accurate profit of churn prediction considering the variable effects of the different promotion offers and customers' possible reactions regarding these offers. In churn prediction calculating the profit of the model is more complex than other classification applications because the total profit depends on the reaction of the customer. We have analyzed this issue and give the formulation to find the appropriate offers for customers considering the total net profit of the company. This formulation maximizes the total net profit of churn prediction model using each customer's churn probability and profitability (CLV). Then we have formulated this problem for different policies of promotion offer selection. There are two major approaches in offer selection and managers have to decide between them. First is to offer same offer for all of the customers and the second, give variable (finite) types of offers for each customer and personalize the offers. The results show that variable incentive offers maximize the total net profit of the bank while some of the high fixed offers (in our study $10 and $20) have negative net profit (cost) for the bank.

6 Responsibility

1. Ashkan Zakaryazad: Problem Statement, Data Source and Description, and Methodology (Profit Maximization by Incentive Offer Selection), and Conclusion.
2. Taewoon Kong: Data Source and Description, Methodology (Classification by Ensemble Methods), Writing a report by Latex, and Conclusion.

References

A. D. Athanassopoulos. (2000). Customer satisfaction cues to support market segmentation and explain switching behavior. *J. Bus. Res.* **47(3)**, 191-207.

Dumas, M., Van der Aalst, W.M.P., and Ter Hofstede, A.H. (2005). When Customers Are Members: Customer Retention in Paid Membership Contexts. *J. Acad. Mark. Sci.*, **26(1)**, 31-44.

M. Colgate, K. Stewart, and R. Kinsella. (1996). Customer defection: a study of the student market in Ireland. *Int. J. Bank Mark.* **14(3)**, 23-29.

Y. Freund and R. E. Schapire. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences.* **55(1)**, 119-139.

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research.* **11**, 169-198.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine.* **6(3)**, 21-45.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review.* **33(1-2)**, 1-39.

L, Breiman. (1996). Bias, Variance, and arcing classifiers. *Technical Report.*

Z. Zhi-Hua. (2012). Ensemble Methods: Foundations and Algorithms. *Chapman and Hall.* 23.

R.E. Schapire and Y. Singer. (1998). Improved boosting algorithms using confidence-rated predictions. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.*, 80-91.

Y. Freund and R.E. Schapire. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences.* **55(1)**, 119-139.