Improving the Effectiveness of Mailings by Building a Response Model for Inactive Customers

Alexander K. Seewald

Seewald Solutions,
Leitermayergasse 33/24, A-1180 Vienna, Austria
alex@seewald.at

Abstract. GE Money Bank is a niche bank mainly specializing in unsecured personal loans, auto loans and retail sales finance. As main part of our marketing strategy, we send out a large number of postal information mails to our customers. Of these, 20% are event-triggered, e.g. by customers opening new loans, closing existing loans, or having their birthdays. The rest are regular mails to our customer base. Here, we distinguish between active customers who have an active contract with us at the point of the mailing, and inactive customers who have not. In this paper, we describe an approach yielding a model that predicts inactive customer mailing response, allowing us to get 30% more responses out of the same number of inactive customers mailed, thus optimizing our mailings. Furthermore, we investigated the effect of feature subset selection on model performance. The model has been deployed and found to perform significantly better than the random sampling approach used previously.

Keywords: Response Model, Machine Learning, Hidden Naive Bayes, Naive-Bayes

1 Introduction

GE Money Bank is a subsidiary of General Electric and part of GE Consumer Finance, a set of banks operating world-wide. We are a niche bank specizializing mainly in unsecured loans to non-selfemployed customers. We have special products for cash loans, auto loans and retail sales finance. Our major product is the cash loan.

For this risky business, we need to maintain high-quality customer data to determine eligible customers and appropriate pricing, which incidentially yields a high-quality data warehouse containing ample information on present and past customers.

As main part of our marketing strategy, every month a large number of information mails are sent out to our customers via the postal service. 20% of these are based on customer-driven events, e.g. opening new loans, closing existing loans, or birthdays, and trigger standardized mailing campaigns every month. The remainder are mails to our mailable customer base, which we mail on average every two months with distinct seasonal mailing campaigns. A set of selection criteria ensures that we only mail customers with at least some potential and interest, e.g. who have not asked for a mailing stop, who have not applied for or received a loan recently, who are not in arrears and so on. For these mailing campaigns, we distinguish between active customers who have an active contract with us at the timepoint of mailing selection, and inactive customers whose last active contract was closed at least four months in the past.

Previously, inactive customers to be mailed were selected by random sampling. However, it would be far more effective to select a same-sized sample of customers by response likelihood, provided an adequate model can be found. With this in mind we started work on a response model for inactive customers in mid-2006, and began to validate mailing response measurements as well as collecting the necessary training data.

The validation of mailing response measurement is of major importance for such work, because without a reasonably precise target variable which indicates on a customer-by-customer level whether or not the response of the customer can be attributed to a specific mailing, there is always the danger of the model learning the idiosyncrasies of the reporting rather than an approximate model of true response. We validated the response measurement in two ways: by asking our customers why they came to us, and by determining a linear model that explains the number of applications per week by a weighted sum of mailings sent out, with some simple assumptions on mailing effect over time[1]. The estimated effect sizes from both approaches are consistent with our mailing response measurement of counting responses of the mailed customers within 60 days from mailing dispatch date.

Therefore, we have used the most current mailing response measurement for determining the true class of training and test samples, with one minor difference: when responses could be assigned to several mailings, the reporting prefers the earliest one (so that mailing performance does not change retroactively) while during training we preferred the latest mailing (which is likely to have a higher impact on the customers mind-set)[2] We chose to train a response model based on incoming applications and not on signed contracts for two reasons: For once, the time from application to contract usually is longer than one week, and its duration can vary greatly, which blurs the exact timing of each response. For twice, the process from application to contract is managed by the risk department

[1] Mailing effect starts in week after mailing dispatch date where it is linearly proportional to the number of mailings sent out, and falls by a multiplicative factor of 1.5 every consecutive week until week 6, after which it is considered zero. Factor 1.5 was estimated from training data. Different initial linear factors were estimated for each mailing type via robust linear regression. Correlation coefficient was 0.74, based on 12 months of weekly data

[2] However, the ROC curves are almost identical using either variant. Only 5.6% of responses can be assigned to more than one mailing, so the effect might have been expected to be quite small.

and we did not want to overburden the model by training it on two very different things at once.

Another major concern was that while we have very good data on currently active customers – driven by the neccessity for good risk scoring – the data on inactive customers is not as recent and in some cases quite outdated. Also, while ongoing payment behaviour for active contracts ensures at least some variance in derived attributes (e.g. paying score), most inactive customers have practically the same attribute values until the day they first respond to one of our mailings – which is exactly what we are trying to predict! So we have also considered features that change from mailing to mailing. A set of features from previous attempts at modeling attrition were also used, and during some unrelated analyses we found other intriguing features – e.g. customers with mobile phone numbers are about twice as likely to respond than those with fixed line phone numbers – which were also added. A full set of features is shown in Table 1.

Rather than using scorecard modelling with logistic regression using proprietary tools and a lot of manual finetuning, we opted for a straightforward machine learning approach using the Open Source Data Mining suite WEKA[4], recently integrated into the Pentaho Business Intelligence suite. WEKA offers a comprehensive set of state-of-the-art Machine Learning algorithms, and has been used extensively by the author since 2001. It includes his contributions of ensemble learning algorithms (StackingC[17], Grading[18]) and an implementation of the Subsequence String Kernel with Lambda Pruning [16].

WEKA also incorporates a comprehensive set of state-of-the-art autonomous feature subset selection (FSS) methods. As initial results looked promising, we designed a more extensive experiment on the usefulness of automated feature subset selection, which will be a major part of this paper.

2 Related Research

[1] showed that neural networks can be used to increase response rates by several magnitudes for relatively small customer segments in marketing. Similar positive results were reported by [13] for the banking sector.

However, [3] and [8] have found that the difference between artificial neural networks (ANN) and multinomial logit models (MNL) is not significant.

[14] present a meta-analysis of a competition, where 44 entries were contributed by 33 individual participants. The topic was churn prediction, which has some similarity to our task: a low number of responses, high number of features and abundant data. Similar to us, they also used downsampling to create a training dataset with uniform class distributions. They found that using Logistic regression and tree-based models correlated with positive results and thus that learning algorithms make a significant difference. However, the overlap of learning algorithms between our approach and theirs is rather small and only includes Logistic Regression. They did not systematically check feature subset selection but found that stepwise selection methods perform best with Logistic regression, which is compatible with our findings. That good models have stay-

ing power is also something we found as the performance of our final model over several months of test mailings was very similar. Their Gini coefficient is related to area-under-ROC-curve, but just measures the area between ROC and random performance, thus $AUC \approx Gini + 0.5$. Thus the best models which they present have similar performance to our best model as measured by Gini, but our model has better lift at 10% recall (3 vs. 2.14), which may reflect the different nature of these two tasks.

[7] propose an information fusion approach to combine data from surveys with historical customer data. This general approach could also be used to augment our data, which is an interesting venue for future research. They claim that the development of data mining has lagged behind the development of tools for collecting and storing the data. This is not the case: there are many learning algorithms capable of processing large amounts of data, and subsampling works very well for the more complex algorithms where this is not feasible.

[6] propose an approach to evaluate prospects for cross-selling in financial services. This is similar to our approach in that both estimate the likelihood of customer response albeit in two different contexts. They use logistic regression as the main learning algorithm, which is a widespread choice in response modelling.[3] However, only ownership of financial services as binary variables are considered as features, while our feature set is richer both in the types of variables and in their contents.

[19] seems to be the only work which also combines customer response modelling with feature subset selection. They propose a wrapper-based feature subset selection based on genetic algorithms and validate on a publicly available dataset. The used dataset is not as unbiased as ours at around 20% for the minority class (vs. 1% here), and their genetic algorithm is a far less systematic way to search through the attribute space than *WrapperSubsetEval*, so we would bound to expect that the latter performs competitively. Sadly, there is no freely available implementation, so we could not check this.

3 Experimental Setup

Here we will describe the background, data collection, training / testing setup and other technical details on our approach.

3.1 Local DWH System

The local datawarehouse runs on a Debian Linux system, 2.4 series kernel, with eight processors and 16 gigabytes of main memory running SAS 9.1, and includes several hundred gigabytes of customer and contract data going back to the very beginning of the bank. Current data is transferred into the DWH once a day, and arrives with 1-2 days delay from the frontend systems. Several important tables are stored as monthly snapshots and thus provide historical information back to about 2001.

[3] A. Kincses, pers. comm.

However, IT systems have changed several times during this time period. Past data has been imported from previous systems and is not fully consistent with present data. Mailing selection was not standardized until late 2004 and not well documented until April 2006. Therefore we have focussed on data from July 2005 onward. This starting point neatly coincides with a major change in the local web-based system used for customer relationship management, where a campaign management tool to provide and manage contact points was added.

We used SAS for constructing features as previous code was to some extent available, and as all the DWH data was available as SAS tables. The feature construction codes were run remotely on the DWH server within a reasonable timeframe[4]. Due to licensing costs, only SAS/Base was available on the server, which precluded the use of the logistic regression package.[5] Therefore, the data was downloaded to a local machine in tab-separated format, run through a Perl program to convert to WEKA's ARFF file format, and afterwards run through WEKA locally.[6] It would have been feasible to run WEKA directly on the DWH server, but it was not possible to have Java installed on as the server is a production system and the set of installed software is therefore strictly limited.

3.2 Data Collection

We used historical data on mailing performance from July 2005 to March 2006. There were several hundred of thousand past mailings with around 1% positive responses. Consistent with our unified mailing performance measurement, we defined positive responses as customers who put in at least one application within 60 days of the mailing dispatch date. In case we could link multiple mailings to the same customer response, we assigned the response only to the most recent one. We cross-checked dispatch dates of all mailings within this period with the bills received by the postal service, and corrected discrepancies manually.

Additionally, we used five inactive customer mailings after the training period – *big_2006-4-7*, *big_2006-5-5*, *big_2006-6-6*, *big_2006-7-3* and *big_2006-7-19* – as independent test sets. The model was deployed in September 2006 as mailing *big_2006-9-4*.

As benchmark to compare the deployed model's performance, we computed the 95% confidence intervals of net conversion rate ($= \frac{signed\_contracts}{customers\_mailed}$) for all eight mailings in 2006, excluding the best- and worst-performing mailing for robustness. Note that as we have trained only a response model, this is a much harder test for the final model.

---

[4] Around 10ms per customer entry.

[5] In a more challenging analysis on a dataset with 100,000 rows and several dozens of attributes, we found the SAS logistic regression lacking: even after 18h, it was still running, obviously having crashed. The WEKA implementation took less than a minute on the same dataset.

[6] Scoring all customers with the final model took several seconds.

3.3 Feature Set

One of the challenges of data collection was that data on inactive customers may be quite out of date – e.g. income, worst payment behaviour and even no. of dependent partners are likely to have changed over the up to four years since these customers last had an active contract. Therefore, we included not only features derived from latest-known customer information, but also more dynamic features such as the number of mailings received by the customer in the previous six months.[7] We are mailing up to six years inactive customers right now and preliminary results indicate that the final model is about equally good on this test group, so this approach seems to have worked.

We reused practically all features from a previous attempt of the marketing department to build an attrition scorecard, and from other scorecards in whose development marketing had previously been involved. Also, previous research has pointed out intriguing features such as the type of telephone number – mobile or fixed line – which we also included. Table 1 shows the full feature set.

3.4 Training/Testing Setup

Because of the biased class distribution – 1% positive, 99% negative responses – we chose to radically downsample our data, creating a *training set* with 50% positive and 50% negative responses[8] and a biased *test set* with 0.5% positive and 99.5% negative responses, using half the positive examples for training and the other half for testing. Additionally, we chose five mailings from after the training set period for which the response was already known, and used them for the evaluation as well.

Suitable learning algorithms include those who return sufficiently diverse class probability distributions. We chose *Naive Bayes* (*NB* [9]), a more complex variant called *Hidden Naive Bayes* (*HNB* [20]), and *Logistic regression* via ridge estimator (*Log* [2]), as implemented in WEKA. *Log* is very similar to a linear Support Vector Machine (SVM, [15]) – in fact most SVM implementation use a logistic model on the SVM outputs when estimating probabilities – so we have chosen not to include SVM on its own. We chose to systematically evaluate the upcoming feature subset selection (FSS) methods to investigate the following three hypotheses.

1. whether FSS has a significant effect vs. the trivial approach of using all features
2. whether wrapper-based FSS – which searches for an optimal subset of features for a given classifier by a large number of runs on the training data – is superior to more basic FSS as is sometimes claimed
3. whether there is a clear winner among the FSS systems.

[7] In fact, a model based only on this single factor achieves an AUC of 0.535 – clearly better than random sampling.

[8] Data available upon request for noncommercial uses. Please mail `alex@seewald.at` stating your name, institution (if any), and purpose of your research project.

Name	Type	Description
Mailings	numeric	no. mailings, last 180 days
TotalPastAppl	numeric	no. applications, last 180 days
PosApps	numeric	no. positively decided applications, last 180 days
NegApps	numeric	no. negatively decided applications, last 180 days
TotalPastVertr	numeric	no. active contracts, last 180 days
TotalPastStorno	numeric	no. of stornos, last 180 days
Inaktiv_in_semiyears	numeric	how long inactive (in semiyears)?
Herkunft	nominal	source of last sold contract (centralized, branch, other)
Produktgruppe	nominal	product of most recent contract
Kundengruppe	nominal	group (active, inactive – always inactive here)
Marital_status	nominal	marital status (married, single, divorced, ..)
Comm_workout_ref	nominal	commercial workout referent (indicates delinquency)
Overall_balance_c	numeric	customer obligo
Overall_balance_k	numeric	company obligo
Dependants	numeric	no. of dependents
Dependant_partner	nominal	spouse with / without income
Days_since_latest_reminder	numeric	days since latest reminder
Amt_past_due_current_balance	numeric	amount past due on current balance
Amt_past_due_instalments	numeric	amount past due on installments
No_accounts	numeric	total no. of past contracts
No_accounts_not_liquidated	numeric	total no. of past contracts which are currently not liquidated (always 0)
Months_since_opendate_oldest	numeric	months-on-book of oldest contract
Total_arrears_amt	numeric	total arrears amount
Total_first_reminder	numeric	no. of first reminders over all contracts
Total_second_reminder	numeric	no. of second reminders over all contracts
Total_third_reminder	numeric	no. of third reminders over all contracts
Worst_payment	numeric	worst paying score over all contracts
Worst_payment_not_liquidated	numeric	worst paying score over non-liquidated contracts
No_accepts	numeric	total no. of positively decided applications
No_deferrals_not_liquidated	numeric	no. of deferrals on all active contracts
Months_since_last_deferral	numeric	months since latest deferral
Highest_days_past_due	numeric	maximum number of days past due
Highest_arrears_ratio	numeric	highest ratio arrears by net balance
Total_plansaldo	numeric	planned net balance over all contracts
Level_latest_reminder	numeric	level of latest reminder
Total_frame	numeric	sum of frame amount over all contracts
Present_ORV	numeric	present outstanding balance
Blank_ORV	numeric	blank portion of outstanding balance
Employed_months	numeric	no. of months working at current employer
Industry	nominal	Industrial area of customer's job
Zip_code	nominal	First two digits of zip code (coarse sociodemographic information)
Net_Income	numeric	Latest known net income of customer
Written_Prove_Salary_Available	nominal	Net income proven by written receipt?
Emp_Type	nominal	Type of employment
Customer_age	numeric	Age of customer (in years)
Bundesland	nominal	Federal district within Austria for customer home address
Tel_Type	nominal	type of telephone (fixed-line, mobile phone, none)
Reminder_Status	nominal	Status of latest reminder
Agreement_status2	nominal	early/late collections
Collections_Stop_Code	nominal	was collection stopped? (=delinquent), and status
Paying_Score	nominal	paying score of most recent contract
Deferal_Status	nominal	customer currently defers payment?
ContractTerm	numeric	term of latest contract (in months)
MOB	numeric	months-on-book of most recent (closed) contract
Loantermcov	numeric	months-on-book by contract-term
PerturnLocal	numeric	local balance by turnover
PerturnNet	numeric	net balance by turnover
Maxoverppl	numeric	maximum arrears, last three months of previous six
OverppObs6	numeric	overpayment by installment, six months ago
Numopt6	numeric	no. of overpayments, last six months
Recency	numeric	last overpayment: how long ago? (months $<= 6$)
Numthrice6	numeric	no. of overpayments $>$ 300 EUR, last six months, monthly payments
Numpay6	numeric	no. of payments, last six months
NumNotpay6	numeric	no. of non-payments, last six months
MeanOverpaymentf3	numeric	mean overpayment, first three months of previous six
MeanOverpaymentl3	numeric	mean overpayment of last three months
MeanOverpayment6	numeric	mean overpayment of last six months
MeanOverpaymentThrend	numeric	trend overpayment first three vs. last three months of previous six
PerBalancePaid6	numeric	paid per installment, last six months
PerBalancePaidl3	numeric	paid per installment, last three months
MaxOverppf3	numeric	maximum overpayment by installment, first three months of previous six
MaxOverppl3	numeric	prop. overpayment by installment, maximum last three months
MaxOverpp6	numeric	prop. overpayment by installment, maximum last six months
m	numeric	month of application (for seasonality)

Table 1. Full feature set. All features computed at resp. mailing dispatch date from historical data.

As we shall shortly see, the answers are: No, No and Yes[9]. We chose practically all supervised FSS systems within WEKA (see weka.attributeSelection), divided into three categories.

[9] Additional caveat: Even with six test mailings, there are almost no significant differences between the winner and all other FSS systems at 95% significance level.

- **Subset-based methods**, which systematically evaluate subsets of attributes. We used Search method BestFirst with $-N$ 1000 (meaning that search stops when 1000 nodes were found which cannot be improved in one step). The methods were *CfsSubsetEval* [5] and *ConsistencySubsetEval* [12].
- **Feature-based methods**, which evaluate only the merit of single attributes. While these are quite fast, they ignore mutual dependency between attributes. We chose to retain the top 10 attributes, since this was about the number of features retained from the subset-based methods in the last step, and also by the final model chosen for deployment. The methods were *GainRatioAttributeEval* (information gain ratio w.r.t. class), *ChiSquaredAttributeEval* (χ^2 statistic with class-based discretization of numeric attributes), *InfoGainAttributeEval* (information gain w.r.t. class), *ReliefFAttributeEval* [11] and *SymmetricalUncertAttributeEval* (symmetrical uncertainty w.r.t. class).
- **WrapperSubsetEval** [10], a costly attribute subset evaluator, with BestFirst $-N$ 1000. This method optimizes the feature set for the given learning algorithm by systematically searching the space of possibly attribute subsets and testing each one with the given classifier. This can obviously lead to overfitting, but gives a feature set that is optimally suited for the given classifier.

As accuracy and error rate are not meaningful for datasets with very biased class distributions, we chose to use *area under the ROC curve* (AUC) as single evaluation measure. 0.5 means performance similar to random selection, 1.0 would mean a perfect ranking with all responders at the beginning when sorted by descending response probability. The FSS methods were run on training data and the selected subsets applied unchanged to all test data as is customary.

4 Results

Here we describe the results of the feature subset selection (FSS) experiments as well as the performance of the final model vs. the previous approach.

4.1 Feature Subset Selection

Table 2 shows the results of all FSS experiments, averaged over all six test sets (*test* and five mailings between training and deployment). For the final model, we chose a combination of *HNB* and *WrapperSubsetEval* because it offered the best performance in terms of AUC on the test set. This may have been a bit premature: Overall, *CfsSubsetEval* would have been the best choice, and *Log* would have been as good as HNB with this FSS method.

It should be noted that with 95% significance level, almost all FSS methods are statistically indistinguishable from *CfsSubsetEval*. Since this includes no feature subset selection, we conclude that FSS has no significant effect on the outcome in terms of AUC and therefore hypothesis 1 & 2 are disproved for lack

Class.		No FSS	Cfs	Cons.	χ^2	GR	IG	ReliefF	SymU	Wrapper
HNB	Avg.	0.750	**0.758**	0.741	0.690	0.635	0.690	0.713	0.677	0.754
	±stD	0.018	0.018	0.016	0.025	0.018	0.025	0.013	0.019	0.018
NB	Avg.	0.719	**0.736**	0.734	0.699	0.649	0.699	0.720	0.691	0.718
	±stD	0.016	0.020	0.017	0.025	0.016	0.025	0.011	0.020	0.016
Log	Avg.	0.751	**0.756**	0.749	0.701	0.650	0.701	0.726	0.694	0.734
	±stD	0.014	0.013	0.015	0.024	0.016	0.024	0.005	0.027	0.015

Table 2. This table shows the results for all feature subset selection experiments, averaged over all test sets. Best performance (non-significant) is shown in **bold**.

of evidence. Hypothesis 3 is confirmed, but not in a significant way – practically all other FSS methods perform as good as *CfsSubsetEval*, except for *GainRatio*.

Figure 1 shows the results for the test set and two of the test mailings. The ROC curves for the other test mailings were very similar and are therefore not shown.

4.2 Final model

The final model was deployed on the 4th of September in the mailing *big_2006-9-4*. The standardized reporting was run two months later. In absolute terms, the *net conversion rate* (NCR, $\frac{signed\_contracts}{customers\_mailed}$) was 0.53% which was significantly better than the previous mailings which used random subsampling.

Overall, the top 25% by descending response probability even had a NCR of 1.01%. The ratio of applications to contracts was relatively stable at around 2 for the top three quarters. Only in the bottom quarter was there a deterioration with roughly four applications for each contract. The basic assumption that a response model yields roughly the same number of contracts as a random selection was thus largely fulfilled. On average, we assume a ratio of roughly 2 for these kinds of mailings.

5 Conclusion

We have described a response model that increases the effectiveness of mailings to inactive customers significantly, allowing us to increase volume by about 30% without increasing mailing costs. Since deployment, the model has been tested and shown to perform significantly better than the previous selection based on random sampling.

We have also investigated feature subset selection methods and found that FSS has no significant effect vs. using all features. In our setting, wrapper-based FSS is not superior to much simpler methods. Overall, *CfsSubsetEval* performs better, albeit not significantly better, than the other methods. So performance concerns cannot be plausibly taken as a motivation for feature subset selection, especially when costly as in the case of *WrapperSubsetEval*.

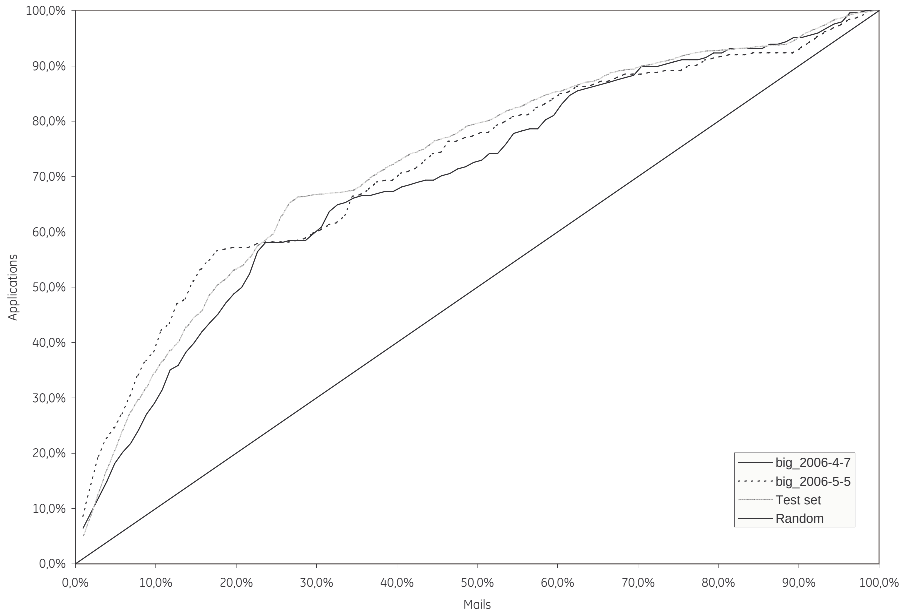

Fig. 1. Inactive customer model: Performance on test set, and two test mailings – about 80% applications at 60% mailings and 60% applications at 30% mailings.

In the future, we plan to add new features to our feature set and investigate feature combination, nominal grouping and numeric interval groupings to see whether results can be further improved. If feasible, we plan to integrate feedback from domain experts in scorecard modelling.

References

1. D. Bounds, D. Ross: Forecasting customer response with neural networks, Handbook of Neural Computation, G6.2, pp. 1-7, 1997.
2. S. le Cessie, J.C. van Houwelingen: Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201, 1992.
3. C.G. Dasgupta, G.S. Dispensa, S. Ghose: Comparing the predictive performance of a neural network model with some traditional market response models, International Journal of Forecasting 10, 235–244, 1994.
4. E. Frank, I.H. Witten: Data Mining - Practical Machine Learning Tools and Techniques (2nd ed). Morgan Kaufmann / Elsevier, 2005. `http://www.cs.waikato.ac.nz/~ml/weka/`
5. M.A. Hall: Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand, 1998.
6. W.A. Kamakura, S. Ramaswami, R. Srivastava: Applying Latent Trait Analysis in the Evaluation of Prospects for Cross-Selling of Financial Services, International Journal of Research in Marketing, 8, 329–349, 1991.

7. W.A. Kamakura, M. Wedel, F. de Rosa, J.A. Mazzon: Cross-selling through Database Marketing: A Mixed Data Factor Analyzer for Data Augmentation and Prediction. International Journal of Research in Marketing 20, 45–65, 2003.
8. A. Kumar, V.R. Rao, H. Soni: An Empirical Comparison of Neural Network and Logistic Regression Models. Marketing Letters, 6, 251–263, 1995.
9. G.H. John, P. Langley: Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338-345. Morgan Kaufmann, San Mateo, 1995.
10. R. Kohavi, G.H. John: Wrappers for feature subset selection. Artificial Intelligence 97(1-2):273-324, 1997.
11. I. Kononenko: Estimating Attributes: Analysis and Extensions of RELIEF. Proceedings of the European Conference on Machine Learning, 171-182, 1994.
12. H. Liu, R. Setiono: A probabilistic Approach to Feature Selection - A Filter Solution. Proceedings of the 13th International Conference on Machine Learning, 319-327, 1996.
13. L. Moutinho, B. Curry, F. Davies, P. Rita: Neural Network in Marketing, New York, Routledge, 1994.
14. S. Neslin, S. Gupta, W. Kamakura, J. Lu, C. Mason: Defection Detection: Improving Predictive Accuracy of Customer Churn Models, Working Paper, Teradata Center at Duke University, 2004.
15. J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning. B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, 1998.
16. A.K. Seewald, F. Kleedorfer: Lambda Pruning - An Approximation of the String Subsequence Kernel. Technical Report, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, TR-2005-13, 2005.
17. A.K. Seewald: How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness. In Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002), Morgan Kaufmann Publishers, San Francisco, 2002.
18. A.K. Seewald, J. Fürnkranz: An Evaluation of Grading Classifiers. In Advances in Intelligent Data Analysis: Proceedings of the 4th International Symposium (IDA-01). Lisbon, Portugal. Springer-Verlag 2001.
19. E. Yu, S. Cho: Constructing Response Model using Ensemble based on Feature Subset Selection. Expert Systems with Applications 30(2) pp. 352-360, February 2006.
20. H. Zhang, L. Jiang, J. Su: Hidden Naive Bayes. Proceedings of the Twentieth National Conference on Artificial Intelligence, 919-924, 2005.