LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR STATISTIK

Credit Scoring and the Optimization concerning Area under the curve

Anne Kraus, Helmut Küchenhoff

Credit Scoring and Credit Control XIII

August 28-30, 2013

Agenda

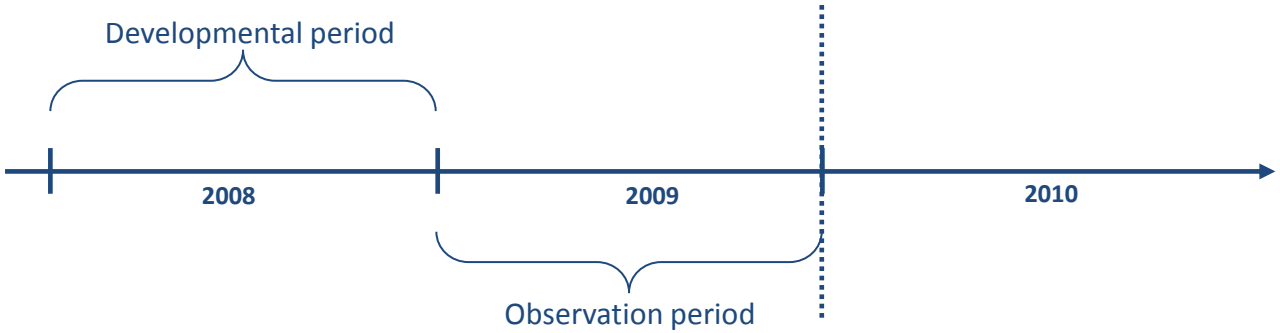
1. Credit Scoring and Data description
2. Classical Method: Logistic Regression
3. AUC approach
4. Boosting
5. Conclusion and Outlook

Credit scoring

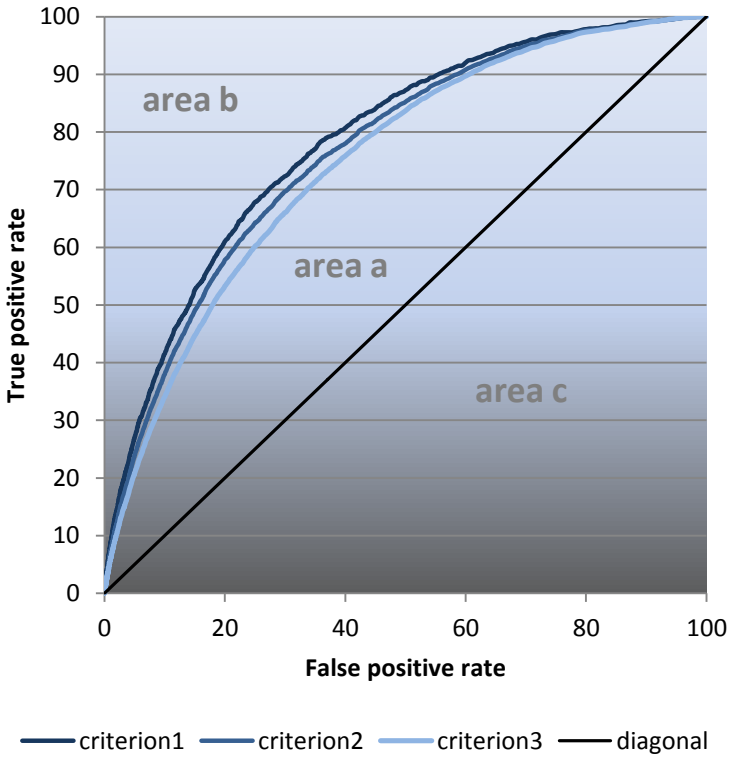
- Credit Scoring models - basis for financial institutions to evaluate the likelihood for credit applicants to default
- Decision whether to grant or reject credit to an applicant
- Data from a German retail bank - specialized on lending consumer credits
- Fully automated application process of this bank
- Credit Scoring model as one part of the whole decision
- Logistic regression as most widely used method for classifying applicants
- Good interpretability is an important advantage of this method
- Benchmark newer methods for optimizing the scoring problem concerning the AUC measure
- Area under the curve (ROC-curve) as measure of performance for prediction accuracy
- AUC as direct optimization criterion

Data description

- Data from a German retail bank
- Large database and long data history
- Application Scoring – 26 attributes regarding process-related limitations
- Variables like credit history, existing accounts, personal information such as age, sex or marital status
- Excluding non representative data
- Different definitions for default
  - Withdrawal of the credit
  - Third dunning letter
  - Time horizon (12 to 30 month)
  - Basel II



Measure of performance

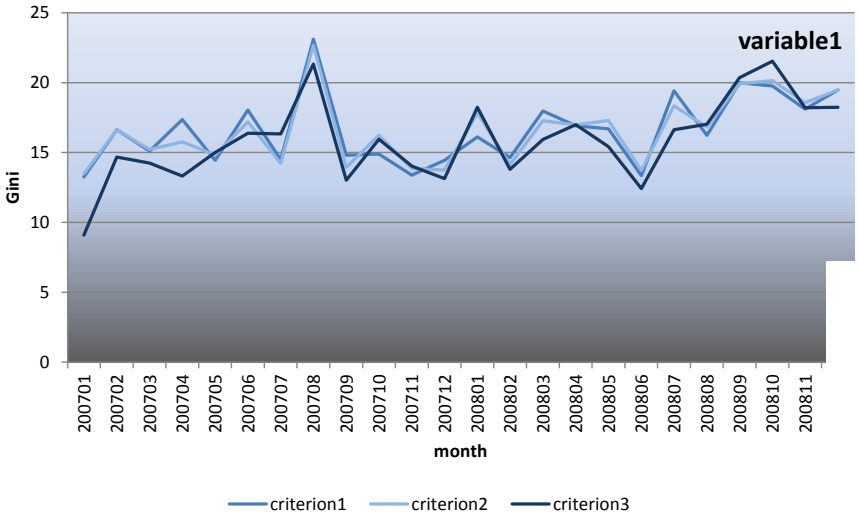


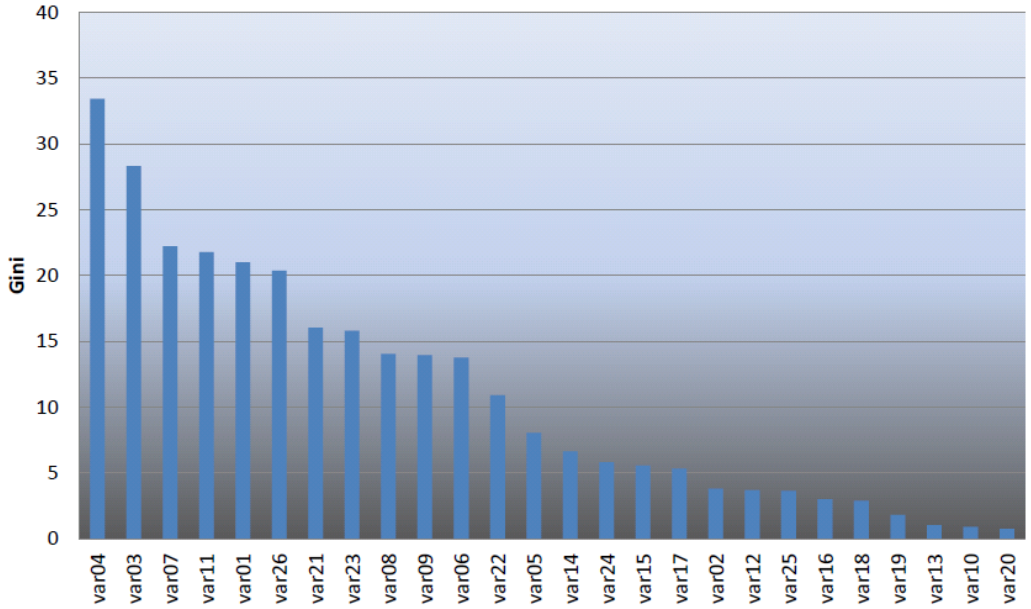
ROC-curve , AUC and Gini coefficient

- Prediciton accuracy
- AUC Area under the curve
  - AUC=area a + area c
- Gini coefficient
  - Gini=area a/(area a + area b)

- Gini=2*AUC-1

Univariate analysis



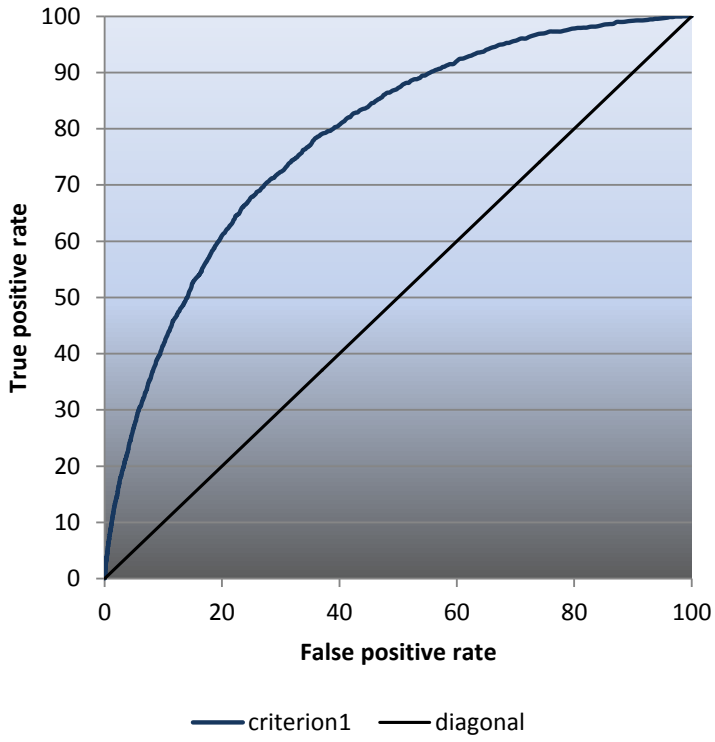


- Univariate Gini coefficient
- Ranking of variables
- Building clusters

Multivariate analysis

- Logistic regression model
  - $\pi_i = P(Y_i = 1|x_i) = G(x_i'\beta)$
  - $G(t) = (1 + \exp(-t))^{-1}$
- Binary outcome variable
  - applicant will default (1)
  - applicant will not default (0)
- Maximum Likelihood estimation
- Test of hundreds of different models
- AUC as measure of performance (compared to AIC)

AUC approach



AUC and Wilcoxon statistic

- AUC Area under the curve
- Wilcoxon statistic (Hanley and McNeil 1982)

$$S(x_{nd}, x_d) = \begin{cases} 1 & \text{if } x_{nd} > x_d \\ 0.5 & \text{if } x_{nd} = x_d \\ 0 & \text{if } x_{nd} < x_d \end{cases}$$

$$\text{AUC} = \frac{1}{n_{nd} \cdot n_d} \sum_1^{n_{nd}} \sum_1^{n_d} S(x_{nd}, x_d)$$

- samples of defaults n_d and non-defaults n_{nd}
- x_d and x_{nd} scores from the defaults and non-defaults

AUC approach - AUC as direct optimization criterion

- Logistic regression with Maximum Likelihood
- Idea of the AUC approach: maximize $AUC(\beta)$

$$\mathrm{AUC}(\beta) = \frac{1}{n_d \cdot n_{nd}} \sum_1^{n_d} \sum_1^{n_{nd}} S(\beta^T(x_d - x_{nd}))$$

$$\hat{\beta}_{AUC} = \arg max_\beta \ \mathrm{AUC}(\beta)$$
$$= \arg max_\beta \frac{1}{n_d \cdot n_{nd}} \sum_1^{n_d} \sum_1^{n_{nd}} S(\beta^T(x_d - x_{nd}))$$

- x_d and x_{nd} scores as vectors of financial indicators
- β vector of coefficient parameters
- $AUC(\beta)$ sum of step functions, non-differentiable with respect to β
- Optimization with Nelder-Mead (1965)

AUC approach

Optimality properties of the AUC approach (Pepe, 2006)

- Exploring linear scores of the form $\mathrm{L}_\beta(x_i) = \mathrm{x}_1 + \beta_2 x_2 + \cdots + \beta_P x_P$
- Neyman-Pearson lemma (1933)
  Rules based on $\mathrm{L}_\beta(x_i) > \mathrm{c}$ are optimal if the score is a monotone increasing function of $\mathrm{L}_\beta(x_i)$
- No other classification rule based on x_i above the ROC curve for $\mathrm{L}_\beta(x_i)$
- Assuming that $\mathrm{L}_\beta(x_i)$ has the best ROC curve, $\mathrm{L}_\beta(x_i)$ has the best ROC curve regarding all linear predictors
- Find coefficients with the best empirical ROC curve
- Since the optimal ROC curve has a maximum AUC value, this measure can be used as objective function to estimate β

- Relationship to the maximum rank correlation estimator of Han (1987)
- Consistency and asymptotically normal distribution (Sherman, 1933)

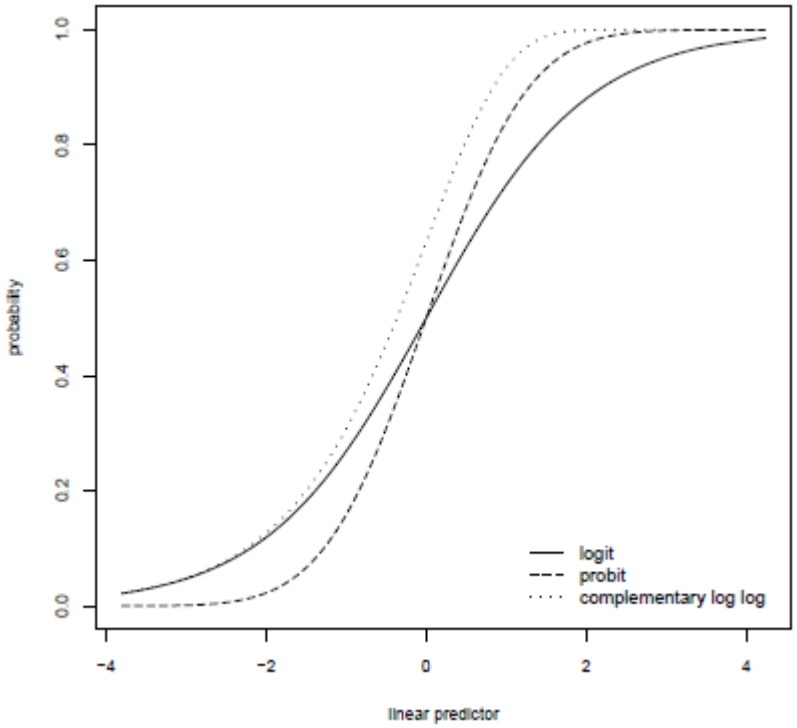
AUC approach

Simulation Study

- Comparison of the logit model and the AUC approach
- Idea: AUC approach outperforms the logit model if link function is not true

- Simulate data with five different link functions
  logit, probit, complementary log log, link1, link2
- Three normal distributed explanatory variables and binary response
  (n=1000, $\beta = (1,0.5,0.3)$)
- 100 simulations
- Estimation on training data and application to validation data

AUC approach

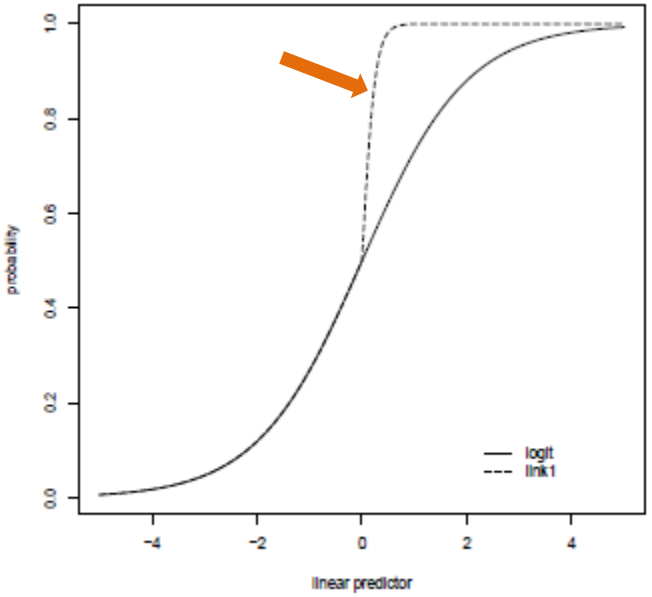


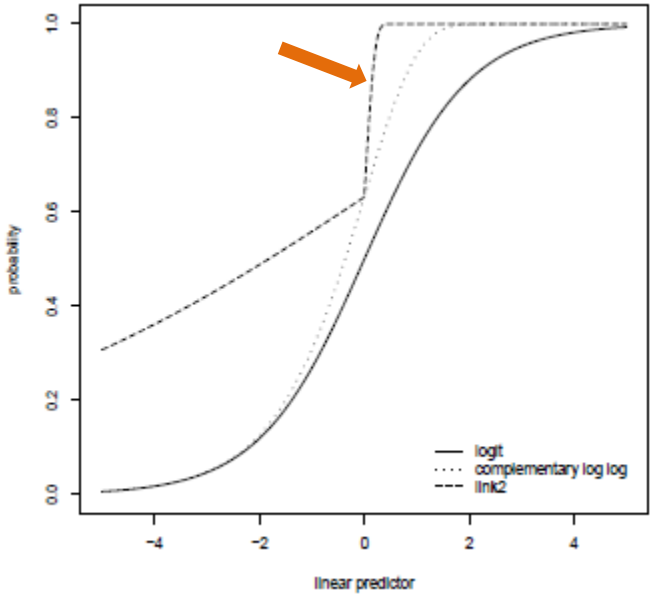
- **logit** response function
  $G(t) = (1 + \exp(-t))^{-1}$
- **probit** response function
  $G(t) = \Phi(t) = \Phi(xi'\beta)$

  with the link function
  $\Phi^{-1}(\pi_i) = x_i'\beta = t$
- **complementary log log** response function
  $G(t) = 1 - \exp(-\exp(t))$

  with the link function
  $\log(-\log(1 - \pi_i)) = x_i'\beta = t$

AUC approach





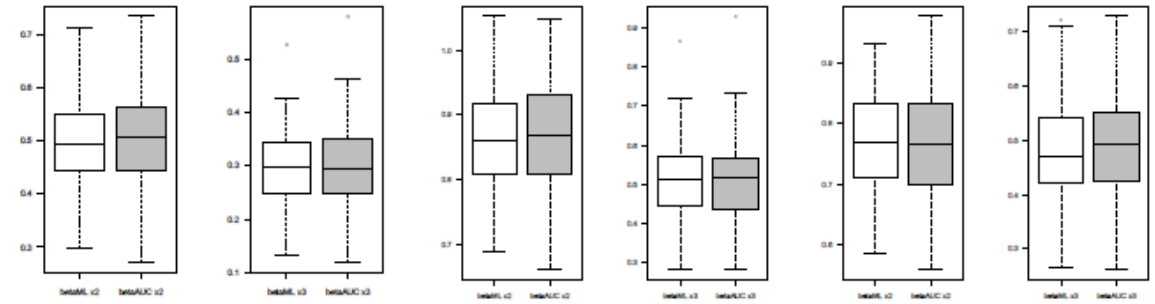
- **link1** (based on the logit link)

$$H(t) = \begin{cases} G(t) & \text{for } t \leq 0 \\ G(8 * t) & \text{for } t > 0 \end{cases}$$

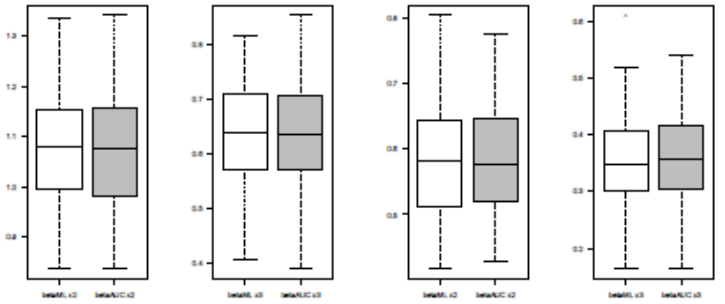
- **link2** (based on the comp log log link)

$$\mathrm{Q}(t) = \begin{cases} G(0.2 * t) & \text{for } t \leq 0 \\ G(5 * t) & \text{for } t > 0 \end{cases}$$

AUC approach



(a) simulated data with logit link (b) simulated data with probit link (c) simulated data with complementary log log link



(d) simulated data with link1 (e) simulated data with link2

- Boxplots to compare the coefficients for the logit model (white) and the coefficients for the AUC approach (grey) based on the 100 simulations

AUC approach

	training				validation			
	logReg		AUC approach		logReg		AUC approach	
	mean	sd	mean	sd	mean	sd	mean	sd
logit	0.7639	0.0181	0.7642	0.0181	0.7633	0.0149	0.7631	0.0148
probit	0.8591	0.0104	0.8593	0.0104	0.8591	0.0111	0.8591	0.0111
comp log log	0.8490	0.0111	0.8493	0.0111	0.8437	0.0120	0.8437	0.0121
link1	0.9016	0.0095	0.9018	0.0095	0.9001	0.0088	0.9002	0.0088
link2	0.8208	0.0130	0.8214	0.0129	0.8214	0.0121	0.8218	0.0121

- Results for the simulation study with 100 simulations
- AUC approach better for link1 and link2

AUC approach

Evaluation on the Credit Scoring data

	training		test		validation	
	LogReg	AUC Opt	LogReg	AUC Opt	LogReg	AUC Opt
2 variables	0.7025	0.7034	0.6650	0.6660	0.6800	0.6804
3 variables	0.7082	0.7098	0.6753	0.6751	0.6951	0.6943
4 variables	0.7104	0.7141	0.6793	0.6801	0.6939	0.6969
5 variables	0.7140	0.7174	0.6835	0.6830	0.6985	0.7002
6 variables	0.7150	0.7183	0.6841	0.6838	0.6993	0.7007
7 variables	0.7344	0.7357	0.7051	0.7048	0.7125	0.7141

- 50% training sample
- One coefficient fixed for normalization reasons
- ML-coefficients basis for the optimization
- No distribution assumption!

Boosting

- Aggregation of many „weak“ classifiers in order to achieve a high classification performance
- Training data with n objects and response $y \in \{+1, -1\}$
- Construct a classification rule $F(x)$
- In each step: find an optimal classifier according to the current distribution of weights on the observation
- Incorrectly classified observations receive more weight in the next iteration (correctly classified observations receive less weight)
- The final model outperforms with high probability in terms of misclassification error rate any individual classifier

- arbitrary loss function exponential $L(y, f) = e^{-yf}$ or logistic $L(y, f) = \log(1 + e^{-yf})$
- η function specifies the type of boosting:
  discrete $\eta(x) = sign(x)$, real $\eta(x) = 0.5\log(\frac{x}{1-x})$ or gentle $\eta(x) = x$
- base learner: classification trees (maxdepth, minsplit)

Discrete AdaBoost

Logistic Regression	defaults	AUC train	AUC test
train050	50%	0,7570	0,7210
train020	20%	0,7610	0,7210
trainORG	2,60%	0,7560	0,7205

- training and test

	AUC	low95%CI	up95%CI	iterations	maxdepth	minsplit
train050	0,6524	0,6426	0,6623	10	1	0
	0,6931	0,6834	0,7028	50	1	0
	0,7050	0,6954	0,7146	100	1	0
	0,7206	0,7111	0,7301	100	2	0
	0,7230	0,7135	0,7325	100	3	0
	0,7247	0,7152	0,7342	100	4	0
	0,7231	0,7137	0,7326	100	5	0
	0,7183	0,7088	0,7278	100	8	0
	0,7095	0,6999	0,7191	100	15	0
	0,7081	0,6985	0,7177	100	30	0
	0,7120	0,7024	0,7215	150	1	0
	0,7222	0,7127	0,7317	150	2	0
	0,7249	0,7154	0,7344	150	3	0
	0,7249	0,7154	0,7344	150	4	0
	0,7258	0,7163	0,7353	150	5	0
	0,7171	0,7076	0,7266	150	8	0
	0,7068	0,6972	0,7164	150	15	0
	0,7044	0,6948	0,7140	150	30	0
train020	**0,7244**	0,7149	0,7339	150	5	0
trainORG	**0,7242**	0,7147	0,7337	150	4	0

- validation

	AUC	LogReg
train050	**0,7313**	0,7275
train020	**0,7293**	0,7280
trainORG	**0,7301**	0,7265

Real AdaBoost

Logistic Regression	defaults	AUC train	**AUC test**
train050	50%	0,7570	0,7210
train020	20%	0,7610	0,7210
trainORG	2,60%	0,7560	0,7205

- training and test

	AUC	low95%CI	up95%CI	iterations	maxdepth	minsplit
train050	0,6610	0,6512	0,6708	150	1	0
	0,6931	0,6834	0,7027	150	2	0
	0,7056	0,6959	0,7152	150	3	0
	0,7120	0,7024	0,7216	150	4	0
	0,7122	0,7026	0,7218	150	5	0
	0,7162	0,7066	0,7257	150	8	0
	0,7166	0,7071	0,7262	150	15	0
	0,7189	0,7094	0,7284	150	15	10
	0,7088	0,6993	0,7184	150	15	100
	0,6606	0,6508	0,6704	150	15	500
train020	**0,7193**	0,7098	0,7288	150	30	0
trainORG	**0,7151**	0,7056	0,7247	150	8	0

- validation

	AUC	low95%CI	up95%CI	iterations	maxdepth	minsplit	**LogReg**
train050	**0,7277**	0,7146	0,7409	150	15	10	**0,7275**
train020	**0,7233**	0,7101	0,7365	150	30	0	**0,7280**
trainORG	**0,7237**	0,7105	0,7369	150	8	0	**0,7265**

Gentle AdaBoost

Logistic Regression	defaults	AUC train	AUC test
train050	50%	0,7570	0,7210
train020	20%	0,7610	0,7210
trainORG	2,60%	0,7560	0,7205

- training and test

	AUC	low95%CI	up95%CI	iterations	maxdepth	minsplit
train050	**0,7251**	0,7156	0,7345	150	5	0
train020	**0,7275**	0,7181	0,7370	150	4	0
trainORG	**0,7262**	0,7167	0,7357	150	30	0

exponential loss

- validation

	AUC	low95%CI	up95%CI	iterations	maxdepth	minsplit	LogReg
train050	**0,7299**	0,7168	0,7431	150	5	0	**0,7275**
train020	**0,7327**	0,7196	0,7458	150	4	0	**0,7280**
trainORG	**0,7295**	0,7163	0,7426	150	30	0	**0,7265**

Boosting AUC

- Component-wise gradient boosting
- Covariates are fitted seperately against the gradient
- Advantage: variable selection during the fitting process
- Different loss functions lead to a huge variety of different boosting algorithms
- In order to optimize the AUC – use the AUC as loss function
- Approximation with sigmoid function for differentiation

- Different base learners in combination with the AUC loss for the Credit Scoring case
  classification trees, linear base learners, smooth effects
- The results for all three base learners outperform the logit model only in a few cases with many included covariates
- In general, the boosting results with AUC loss underachieve compared to the logit model

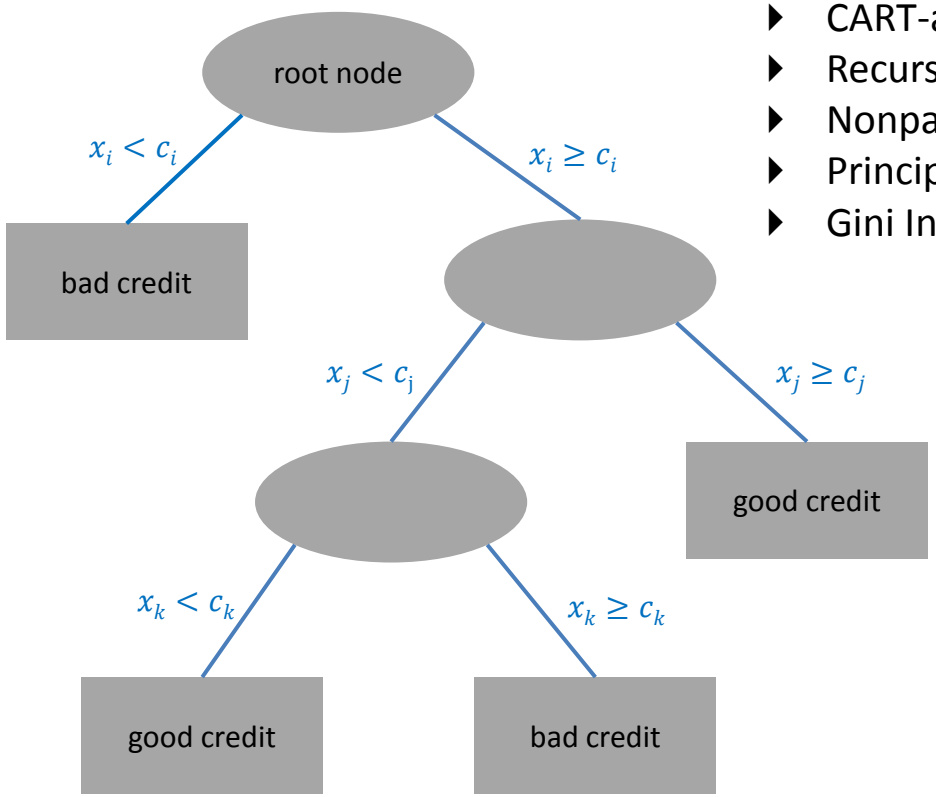
Conclusion and Outlook

- Robust results and good interpretability with logistic regression
- Slight improvements for the AUC approach by direct AUC optimization
- Advantage of the AUC approach: no distribution assumption
- AUC loss behind the expectations
- Outperforming outcomes with boosting algorithms

- Further work with single classification trees, random forests and boosting
- Interpretability versus prediction accuracy
- Interesting results and improvements with newer statistical methods and algorithms from machine learning

BACKUP

Classification trees



- CART-algorithm Breiman et al. (1984)
- Recursive partitioning
- Nonparametric approach
- Principle of impurity reduction
- Gini Index as impurity measure

Random forest

1. n_{tree} bootstrap samples from the learning sample; n_{tree} parameter for number of trees

2. For each generated bootstrap sample grow a classification tree with following aspects:
   - In each node not all predictor variables are used to choose the best split but a special number of randomly selected attributes
   - m_{try} describes this number of randomly preselected splitting variables
   - Random forests grow unpruned trees

3. For predicting new data aggregation of the prediction from the n_{tree} trees; majority votes for classification; relative frequency for all trees

Random forest

variable importance measures

mean decrease in node impurity

- averaged over all trees: total decrease in node impurities from splitting on the variable
- Gini index for classification

mean decrease in accuracy

- measure is computed from permuting OOB data
- recording prediction error on the OOB data
- prediction error after permuting each predictor variable
- the difference between the two are averaged over all trees and normalized by the standard deviation of the differences