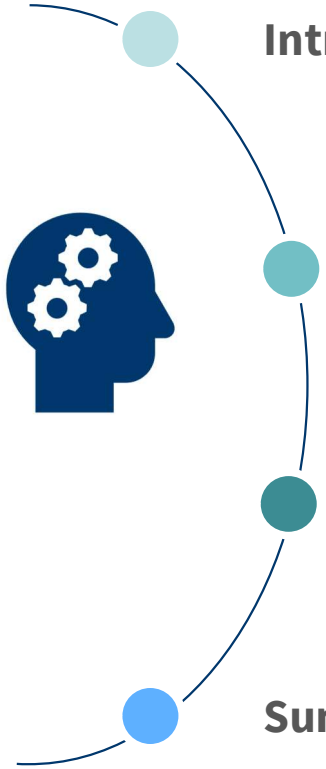
Business Analytics & Data Science

Feature Engineering and Selection

Stefan Lessmann

Agenda



Introduction

Feature Engineering
Common feature transformations, weight-of-evidence, feature extraction

Feature Selection
Motivation, challenges, strategies, filter & wrapper approaches

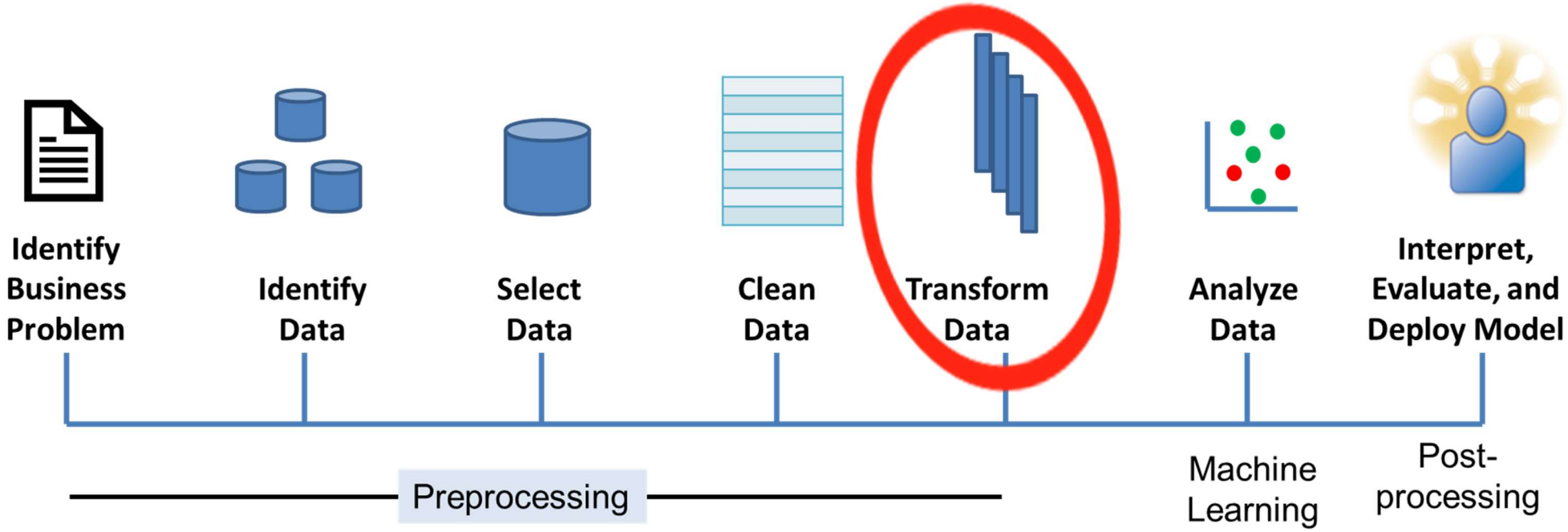
Summary

Introduction

Feature Engineering & Selection in the Analytics Process Model

Activity in the scope of data preparation

- **Typically considered part of the data transformation step**
- **Much interdependency with subsequent process stages**



Feature Engineering

Create informative variables for an analytical model

"Feature Engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning and is both difficult and expensive. … Feature engineering is an informal topic, but it is considered essential in applied machine learning."

- Typically manually performed by a data analyst
- Calculation of new – better – variables
- The best way to improve a predictive model is to obtain better (more informative) features
- To be distinguished from feature (or representation) learning
  - Learning algorithms crafts features as part of the learning process
  - Studied in the scope of deep learning, especially for computer vision and language processing

Based on Wikipedia

Feature Selection

Reduce number of inputs in an analytical model

Feature Selection removes a subset of the variables prior to or in the process of building an analytical model. It is more popular in predictive compared to descriptive analytics because the relevance of a feature is easier to evaluate when a target variable is given. The objectives of feature selection are manifold and include

- Interpretation
  - Parsimonious models are considered easier to understand
  - Consider the number of terms in a regression model
- Reducing costs associated with data collection and storage
- Removing noisy/irrelevant/redundant variables might increase predictive accuracy
  - Curse of dimensionality
  - Multicollinearity
- Accelerates model development and application

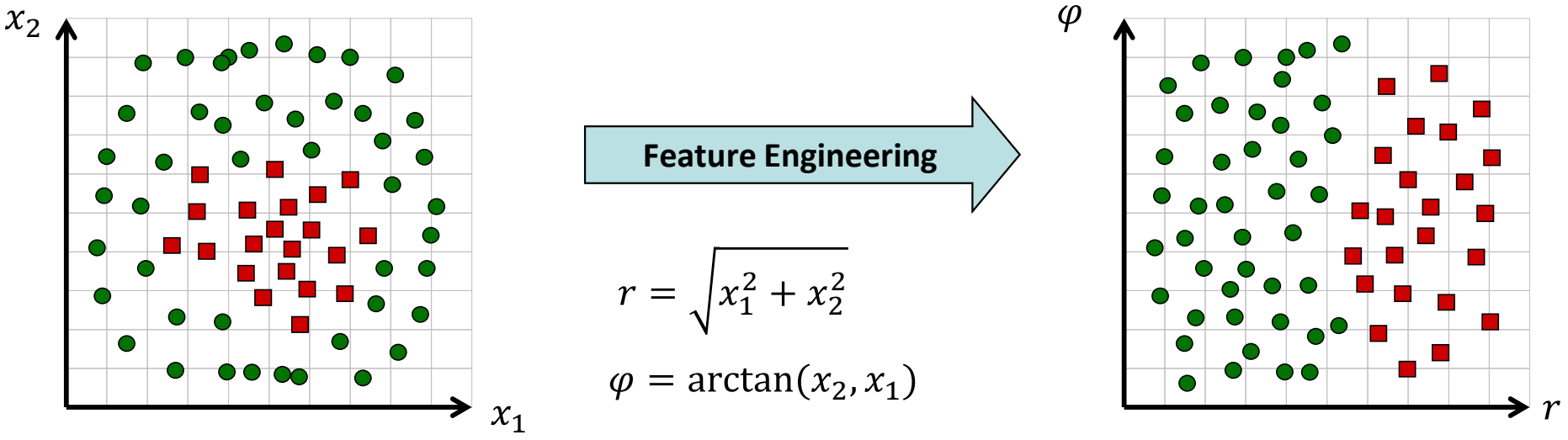
Adapted and extended from Guyon & Elisseeff (2003)

Feature Engineering

Motivation, common transformations, weight-of-evidence, feature extraction

Feature Engineering
Motivating Example

- **Transform available variables into new variables**
  - Linear / nonlinear projection of variables
  - Clever transformation / combination of variables
- **Convert to polar coordinates**

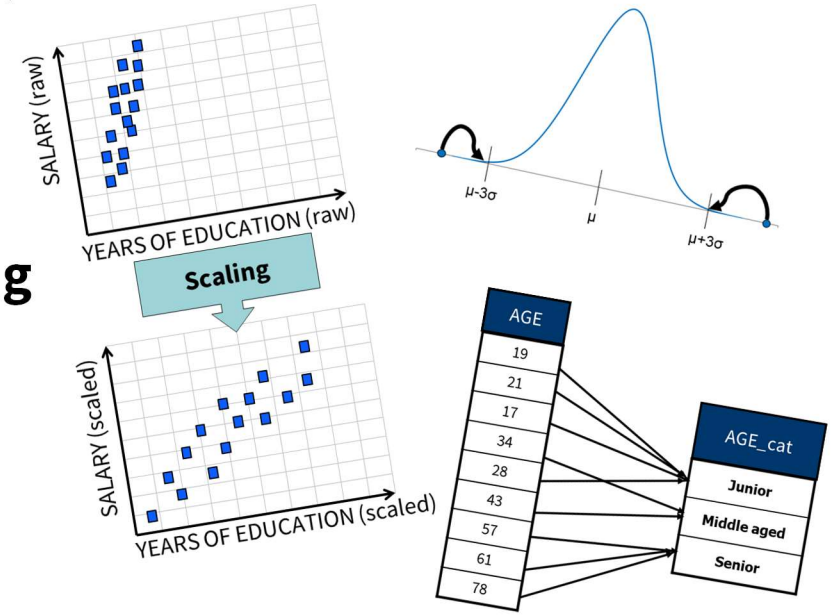


Based on https://www.kdnuggets.com/2018/12/feature-engineering-explained.html

Feature Transformation

Standard operations (covered in chapter 4)

- **Truncation of outliers (z-score, inter-quartile range)**
- **Scaling of numeric variables**
  - Z-transformation
  - Min/Max scaling
- **Categorization of numeric variables and re-grouping**
  - Equal width/frequency binning
  - Supervised discretization (decision trees, chi-square analysis)
- **Coding of categorical variables**



$$x_n = \frac{x_o - \min(x_o)}{max(x_o) - \min(x_o)} \cdot (max_n - min_n) + min_n$$

Common Feature Transformations

- **Ratio variables (very common in finance)**
  - Retail lending: debt-to-income, loan-to-value (of the collateral)
  - Corporate lending: working capital / total assets, market value of equity / book value of total liabilities
- **Aggregation and trend variables**
  - Consider multiple measurements of the same quantity
    - Common for variables that change over time
    - Weekly credit card spending, monthly account balance, quarterly bureau score
  - Aggregation to obtain scalar feature value (min/max/avg, quantiles, …)
  - Trend variables
    - Absolute trend $\left(\frac{x_t - x_{t-i}}{i}\right)$
    - Relative trend $\left(\frac{x_t - x_{t-i}}{x_{t-i}}\right)$

Common Feature Transformations

Transform feature values to increase normality and stabilize variance

- **Logarithmic transformation**

$$x^{(t)} = g(x) = \log(x)$$

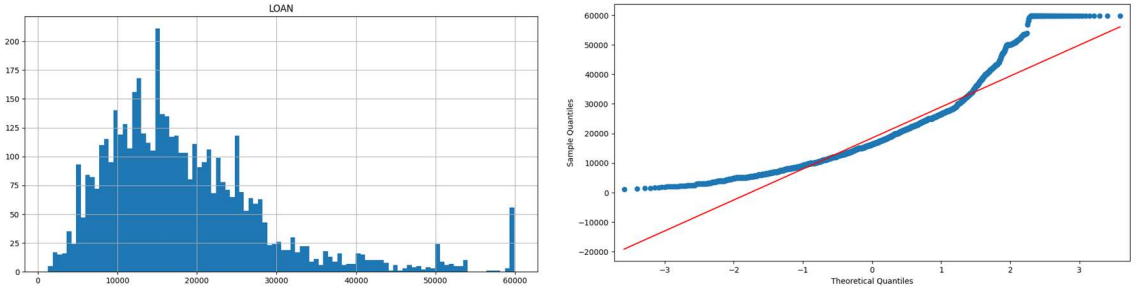
- **Box Cox transformation**

$$x^{(t)} = g(x;\lambda) = \begin{cases} \dfrac{x^{\lambda}-1}{\lambda} & \text{for } \lambda \neq 0 \\ \log(x) & \text{for } \lambda = 0 \end{cases}$$

- **Yeo Johnson Transformation**

$$x^{(t)} = g(x;\lambda) = \begin{cases} ((1+x)^{\lambda}-1)/\lambda & \text{for } \lambda \neq 0, x \geq 0 \\ \log(x+1) & \text{for } \lambda = 0, x \geq 0 \\ -\dfrac{(1-x)^{2-\lambda}-1}{2-\lambda} & \text{for } \lambda \neq 2, x < 0 \\ -\log(-x+1) & \text{for } \lambda = 2, x < 0 \end{cases}$$

- **Shift values if $x \leq 0$**
- **Model selection approach to set λ**



Distribution and QQ plot of feature LOAN in the HMEQ data set before and after transformation

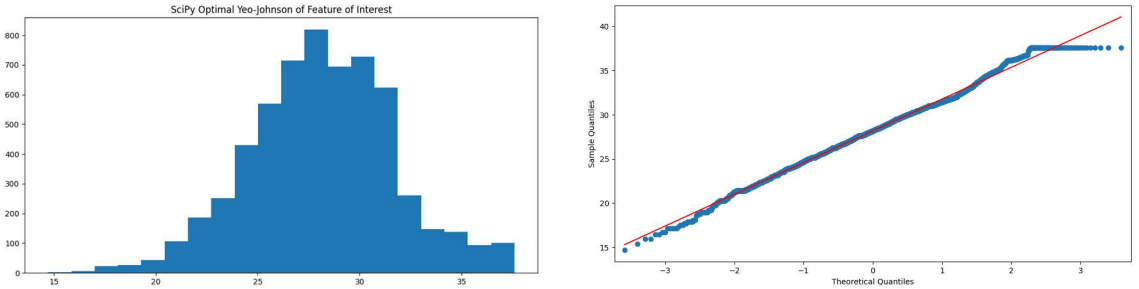


Image source: https://github.com/Humboldt-WI/bads/blob/master/demo_notebooks/9_nb_feature_engineering.ipynb

Feature Transformations for Categorical Variables

Weight of evidence (WOE) coding

- **Replace categorical variable with one numeric variable**
- **Measures association with target variable per category level**
- **Computation**
  - Credit scoring example
  - Two classes of good and bad risks
  - $p(\mathbf{BAD})_{cat}$ is the fraction of bad risks with category level cat
  - $p(\mathbf{GOOD})_{cat}$ is the fraction of good risks with a category level

$$WOE_{cat} = \ln\left(\frac{p(\mathbf{BAD})_{cat}}{p(\mathbf{GOOD})_{cat}}\right)$$

- **Simplification: drop log**
  - Sometimes called supervised ratio
  - See Moeyersoms & Martens (2015) for an empirical comparison

Feature Transformations for Categorical Variables

WOE coding exemplified for a raw variable AGE with seven levels

AGE	Count	Distr. Count	GOODS	Distr. GOODs	BADS	Distr. BADs	WOE
Missing	50	2.50%	42	42/1806=2.33%	8	8/194=4.12%	57.28%
18-22	200	10.00%	152	8.42%	48	24.74%	107.83%
23-26	300	15.00%	246	13.62%	54	27.84%	71.47%
27-29	450	22.50%	405	22.43%	45	23.20%	3.38%
30-35	500	25.00%	475	26.30%	25	12.89%	-71.34%
35-44	350	17.50%	339	18.77%	11	5.67%	-119.71%
44+	150	7.50%	147	8.14%	3	1.55%	-166.08%
Total	**2000**	**100%**	**1806**	**100%**	**194**	**100%**	

WOE = Ln [Distr. BAD / Distr. GOOD] x 100

Ln(4.12/2.33)

Feature Transformations for Categorical Variables

Implementing WOE in practice

- **Estimate WOE score on a separate set of data**
- **Replace category level with corresponding WOE score**
- **Regression example** $y = \beta_0 + \beta_1 \text{AMOUNT} + \beta_2 WOE(\text{AGE})$
- **Cases with the same category level get assigned the same WOE score (e.g. C3 and C8)**
- **No increase in dimensionality**
- **Applicable if cardinality is high** (see Moeyersoms & Martens, 2015)

ID	G/B	AMOUNT	AGE	WOE(AGE)
C1	G	€25,000	Missing	50.05%
C2	G	€15,000	18-22	90.59%
C3	B	€75,000	23-26	61.82%
C4	G	€30,000	27-29	3.05%
C5	B	€67,500	30-35	-66.27%
C5	B	€40,000	35-44	-112.70%
C7	G	€16,000	44+	-157.90%
C8	G	€30,000	23-26	61.82%
C9	B	€55,000	44+	-157.90%
C10	G	€95,000	27-29	3.05%

Feature Transformations for Categorical Variables

Pitfalls when using WOE in practice

- **Sparsely populated levels**
  - Extreme / undefined values due to WOE calculation

$$WOE_{cat} = \ln\left(\frac{p(\mathbf{BAD})_{cat}}{p(\mathbf{GOOD})_{cat}}\right)$$

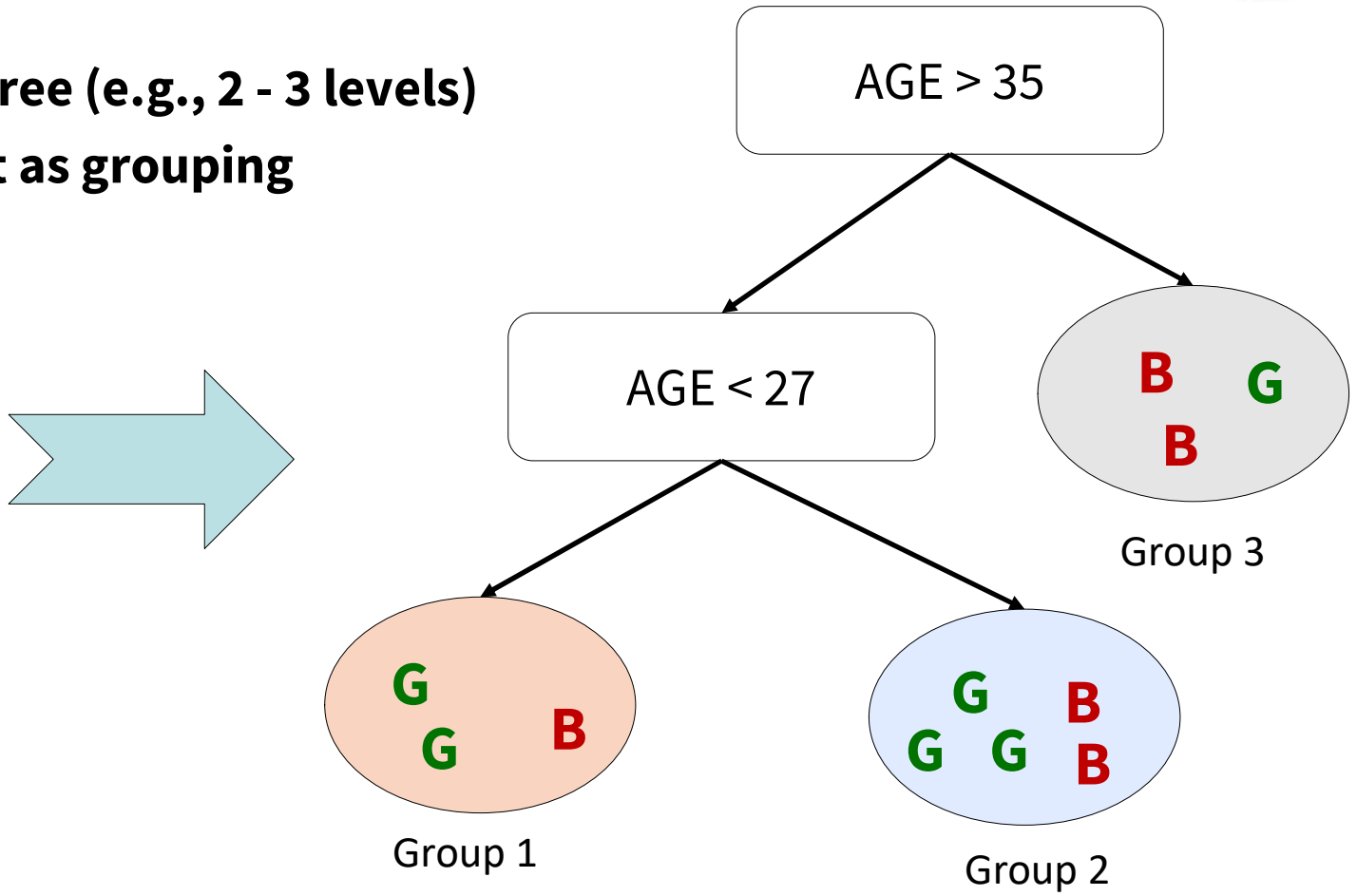
  - Remedy: include artificial data points and adapt WOE calculation (Zdravevski et al., 2011)
- **Novel levels**
  - For example in a hold-out test set
  - Remedy: set WOE = 0 for novel level → average risk (Moeyersoms & Martens, 2015)

Feature Transformations for Categorical Variables

Tree-based projection

- **Create shallow decision tree (e.g., 2 - 3 levels)**
- **Use leaf node assignment as grouping**
- **Univariate example**

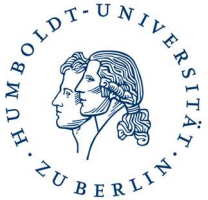
ID	G/B	AMOUNT	AGE
C1	G	€25,000	27-29
C2	G	€15,000	18-22
C3	B	€75,000	23-26
C4	G	€30,000	27-29
C5	B	€67,500	30-35
C5	B	€40,000	35-44
C7	G	€16,000	44+
C8	G	€30,000	23-26
C9	B	€55,000	44+
C10	G	€95,000	27-29



(Niculescu-Mizil et al., 2009)

Feature Transformations for Categorical Variables

Rejoinder

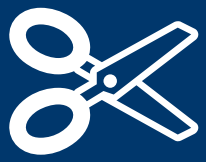


- **We used to encode categorical variables using dummy variable**
- **WOE and tree-based projection avoid increase of dimensionality c.f. dummy coding**
- **They achieve this by using the target variable**
  - Extracting information from the target to find a *better* way to encode the category
  - Also called target (en)coding
  - Only feasible in supervised learning
- **See Python library Category Encoders for further approaches to encode categorical features with or w/o using the target variable**
  - https://contrib.scikit-learn.org/category_encoders/index.html

Feature Engineering Summarized

- **Easy to generate many new features**
  - Multiple aggregation functions
  - Parameters of calculations (e.g., trend over last 2, 3,… month)
  - Tree-based coding for single features, pairs of features, triples of features
- **Feature selection is crucial**
  - Control dimensionality and protect against over-fitting
  - Runtime and memory constraints
  - Also little theory/a priori information which transformation works best

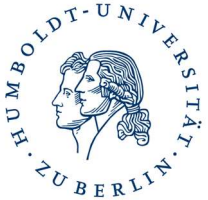
"*The best way to improve a predictive model is to obtain better (more informative) features*"



Feature Selection

Motivation, challenges, strategies, filter & wrapper approaches

Feature Selection

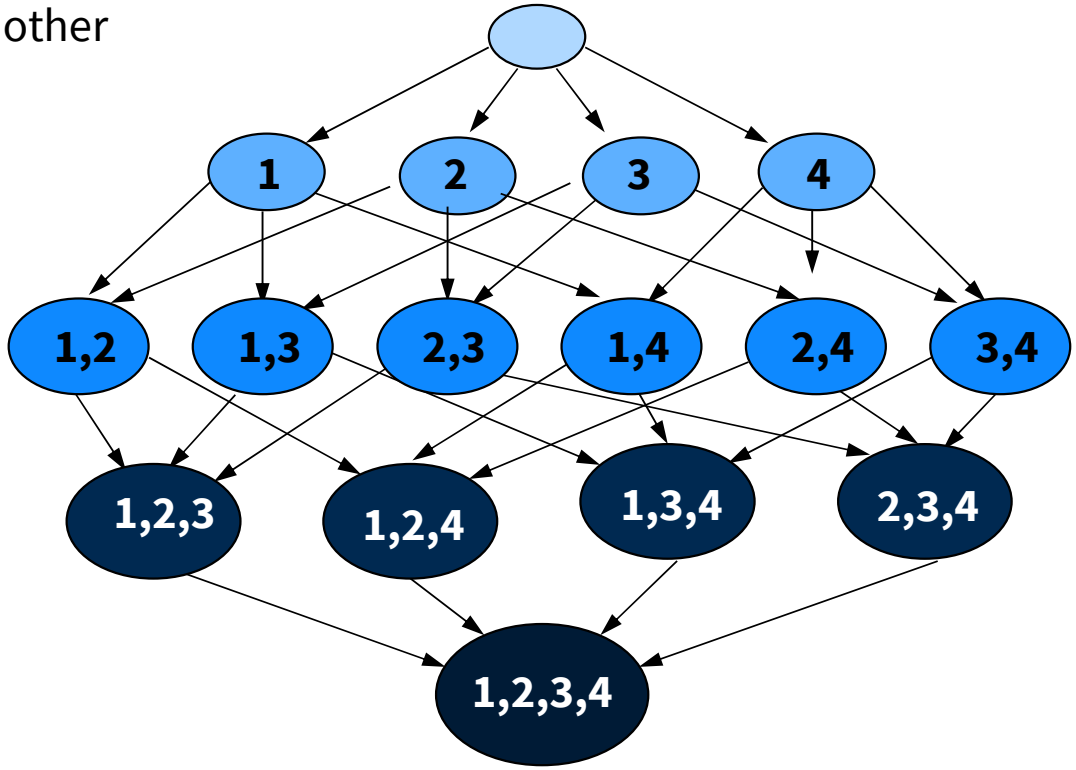


- **Reduce number of inputs in a prediction model**
- **Motivation**
  - Parsimonious models are easier to understand
  - Reduces costs associated with data collection and storage
  - Removing noisy/irrelevant/redundant variables might increase predictive accuracy
    - Curse of dimensionality
    - Multicollinearity
  - Accelerates model development and application
- **Suggested reading: Guyon & Elisseeff (2003)**

Feature Selection Challenges

Feature selection is a combinatorial problem

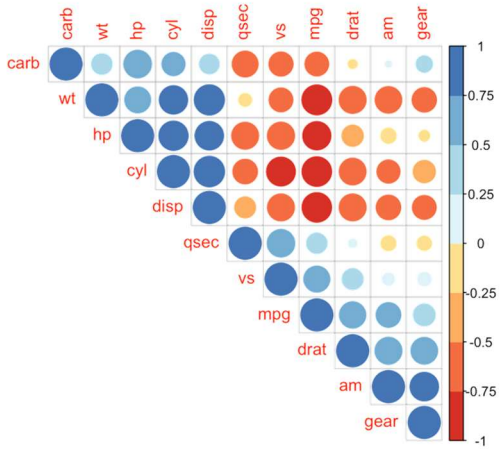
- **Key problem: find best variable subset**
  - Features should be highly correlated with the target
  - Feature should be uncorrelated with each other
- **Combinatorial problem**
  - With n variables: 2^n-1 subsets
  - Approached by means of heuristic search
    - Forward selection
    - Backward elimination
    - Stepwise selection
    - Evolutionary algorithms

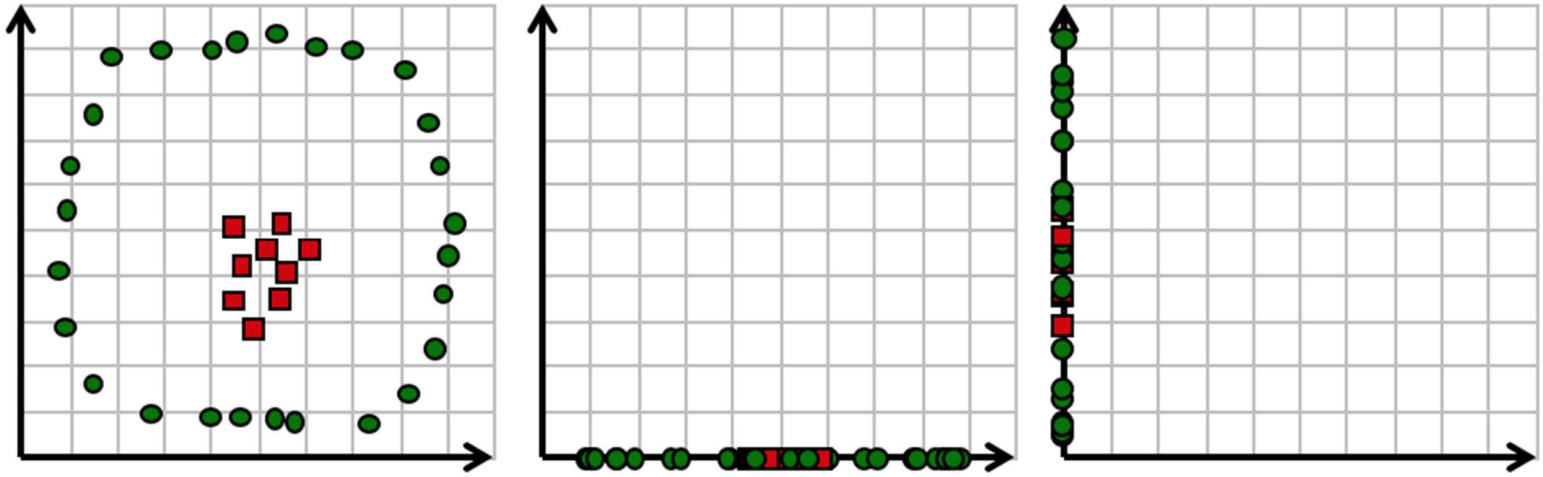


Feature Selection Challenges

Redundancy, relevance, and interaction

- **Features can be relevant but redundant**
  - Example: highly correlated variables
  - Pairwise correlation can reveal redundancy
- **Feature interaction**
  - Variables are predictive when employed together
  - But meaningless when considered individually
- **Many approaches for feature selection operate in a uni-variate manner**





Strategies Toward Feature Selection

Recommended reading: Kohavi & John (1997)

- **Filter approach**
  - Uses statistical indicator
  - Assess one variable at a time
- **Wrapper approach**
  - Uses prediction model
  - Iteratively build & assess model with different variables
  - Examples: forward selection, backward elimination

Filter Methods for Feature Selection

	Continuous target (e.g., sales)	Discrete target (e.g., credit default)
Continuous variable (e.g., income)	Pearson correlation	Fisher score
Categorical variable (e.g., marital status)	Fisher score ANOVA analysis	Chi-square analysis Cramer's V Information value Gain/entropy

Filter Methods for Feature Selection

Pearson Correlation

- **Compute Pearson correlation between each continuous variable and the continuous target variable**
- **Always varies between -1 and +1.**
- **Keep only variables with high correlation**
  - For example $|\rho| > 0.50$
  - Or keep, e.g., top 10%

Filter Methods for Feature Selection

Chi-squared-based filter

- **Discrete target and one categorical variable**
- **Compare empirical to theoretical frequencies**
- **Under the assumption of independence**
  - P(married & good risk) = P(married)*P(good risk) = 0.6 * 0.8
  - Expected number of good risks that are married is 0.6*0.8*1000 = 480

Observed frequencies

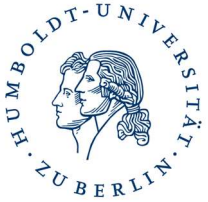
	Good risk	Bad risk	Total
Married	500	100	600
Not Married	300	100	400
Total	**800**	**200**	**1000**

Independence frequencies

	Good risk	Bad risk	Total
Married	480	120	**600**
Not Married	320	80	**400**
Total	**800**	**200**	**1000**

Filter Methods for Feature Selection

Chi-squared-based filter (cont.)



- **For above example:**

$$\chi^2 = \frac{(500-480)^2}{480} + \frac{(100-120)^2}{120} + \frac{(300-320)^2}{320} + \frac{(100-80)^2}{80} = 10.41$$

- **χ^2 is chi-squared distributed with *l*-1 degrees of freedom**
  - with *l* representing the number of category levels
  - The bigger (lower) the value of χ^2, the more (less) predictive the variable
- **Use as filter**
  - Rank all variables with respect to their χ^2 value
  - Select the most predictive variables
- Cramer's $V = \sqrt{\frac{\chi^2}{n \cdot min(l-1, c-1)}}$
  - where n denotes the number of observations, l the number of levels of the categorical feature, and c the number of classes of the target variable (e.g., good vs. bad).
  - Bounded between 0 and 1 with higher (lower) values indicating more (less) predictive variables
  - Threshold of 0.10 is sometimes used as a rule of thumb

Filter Methods for Feature Selection

Fisher score

- **Pairs of categorical and continuous variables**
  - Often used when target is categorical (e.g., binary)
  - Can also be used in regression to assess categorical variables
- **Definition**

$$FS = \frac{|\bar{x}_G - \bar{x}_B|}{\sqrt{s_G^2 + s_B^2}}$$

- **Essentially generalizes to an ANOVA test in the case of multiple categories**

Filter Methods for Feature Selection

Information value (IV)

- **Measure of predictive power**
- **Based on WOE**
  - Useful to select predictive variables
  - Useful to assess suitability of a categorization
- **Rule of thumb**
  - < 0.02 : not predictive
  - 0.02 – 0.1 : weak
  - 0.1 – 0.3 : medium
  - 0.3 + : strong

AGE	Distr. GOOD	Distr. BAD	WOE	IV
Missing	2.33%	4.12%	57.28%	0.0103
18-22	8.42%	24.74%	107.83%	0.1760
23-26	13.62%	27.84%	71.47%	0.1016
27-29	22.43%	23.20%	3.38%	0.0003
30-35	26.30%	12.89%	-71.34%	0.0957
35-44	18.77%	5.67%	-119.71%	0.1568
44+	8.14%	1.55%	-166.08%	0.1095

IV= 0.6502. Thus, AGE is a strong predictor.

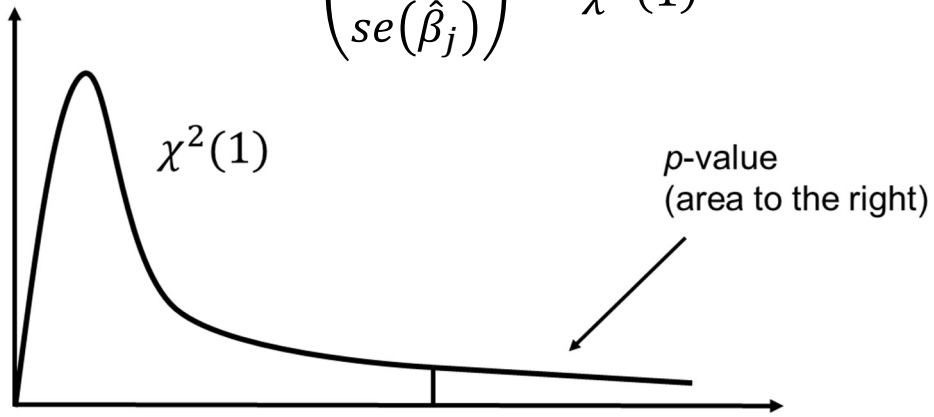
$$IV = \sum_{cat} \left((p(\mathbf{BAD})_{cat} - p(\mathbf{GOOD})_{cat}) \cdot \mathrm{WOE}_{cat} \right)$$

Forward/ Backward/ Stepwise Regression

Setting in conventional multivariate regression

- **Perform hypothesis test to decide upon variable importance**
  - $H_0: \beta_j = 0$ vs. $H_1: \beta_j \neq 0$ (e.g., Wald test in linear regression)
  - Use the *p*-value to decide on variable importance
    - Small (large) *p* means important (unimportant) variable
    - Rule of thumb:
      - $p < 0.01$: highly significant
      - $0.01 < p < 0.05$: significant
      - $0.05 < p < 0.10$: weakly significant
      - $p > 0.1$: not significant

$$\left(\frac{\hat{\beta}_j}{se(\hat{\beta}_j)}\right)^2 \sim \chi^2(1)$$



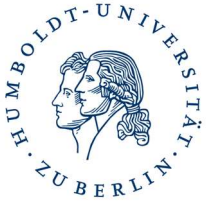
- **Start with empty (full) model and add (remove) one variable at a time**
- **Note that this is an in-sample view on variable importance**

Backward Regression Trimming

Model-agnostic wrapper approach for regression and classification

- **Pseudo-code**
  - Estimate full model with *k* features and estimate its performance on the validation set
  - Estimate *k* models with *k*-1 features and select the one with best validation set performance
  - Estimate *k*-1 models with *k*-2 features and select the one with best validation set performance
  - Continue with these steps until no features are left in the model
  - Chose the overall best model based on validation set performance
  - Estimate a final model from union of training and validation set
- **Based on out-of-sample performance**
  - Assessed in terms of some indicator of predictive accuracy
  - Regression: MSE, MAE, MAPE, … ; Classification: AUC, PCC, F1, …
  - Can also approximate error costs
- **Rigorous but costly**

Filters versus Wrappers



Filter

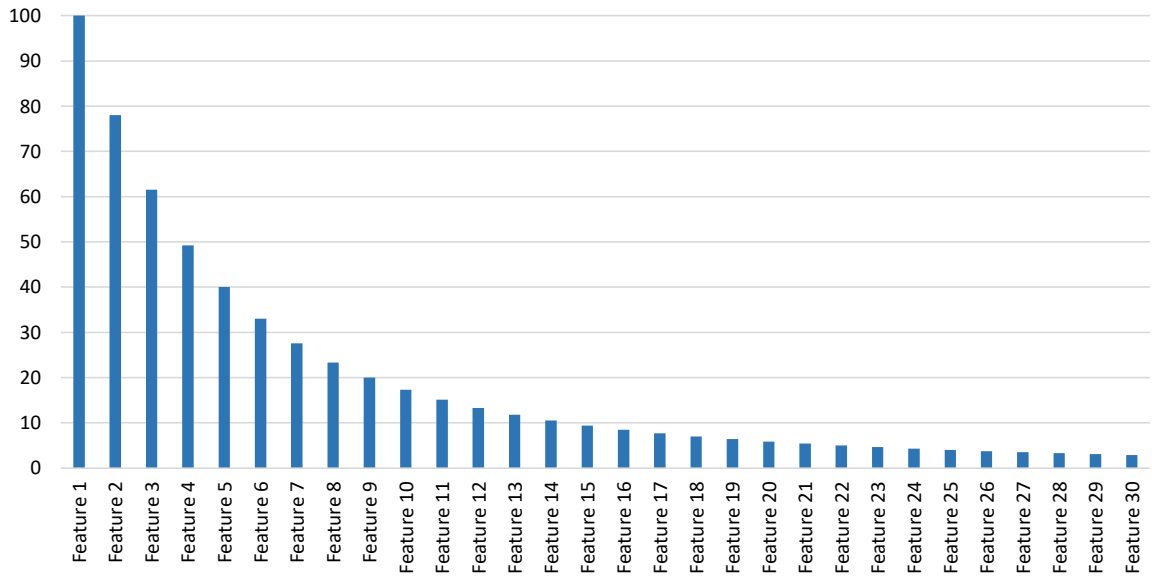
- **Typically less effective**
  - Model agnostic assessment
  - Only statistical indicator
- **Typically used in a univariate way**
  - Redundancy problem
  - Interaction problem
- **Lower computational cost**
- **Different indicators for data sets with numeric & categorical variables**

Wrapper

- **Typically better performance**
- **Multivariate evaluation**
  - Forward/backward/stepwise
  - Able to detect (some) redundancy/interaction
- **Higher computational cost**
- **Requires auxiliary validation data to assess variable subsets**

Filters versus Wrappers
Practical suggestions

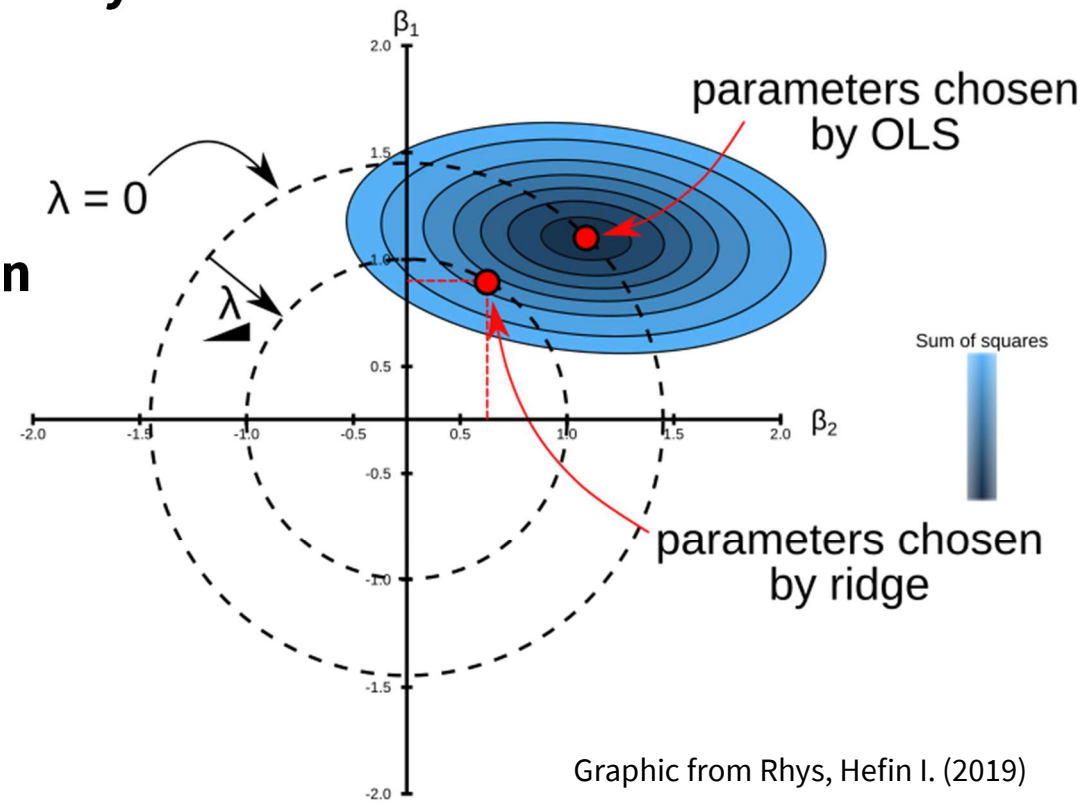
- **Hybrid strategy often useful**
  - Start with filter and remove poor variables
  - Then switch to wrapper for additional selection
- **Criterion for selection needs to be defined**
  - Given ranking of feature importance
    - By correlation, p-value, …
    - Permutation importance (see later)
  - Examine importance distribution and decide on threshold
    - For example correlation ranking
    - Drop feature if $|\rho|$ less than e.g. 0.4
    - Elbow method



Embedded Methods

Feature selection as part of model estimation (i.e., learning)

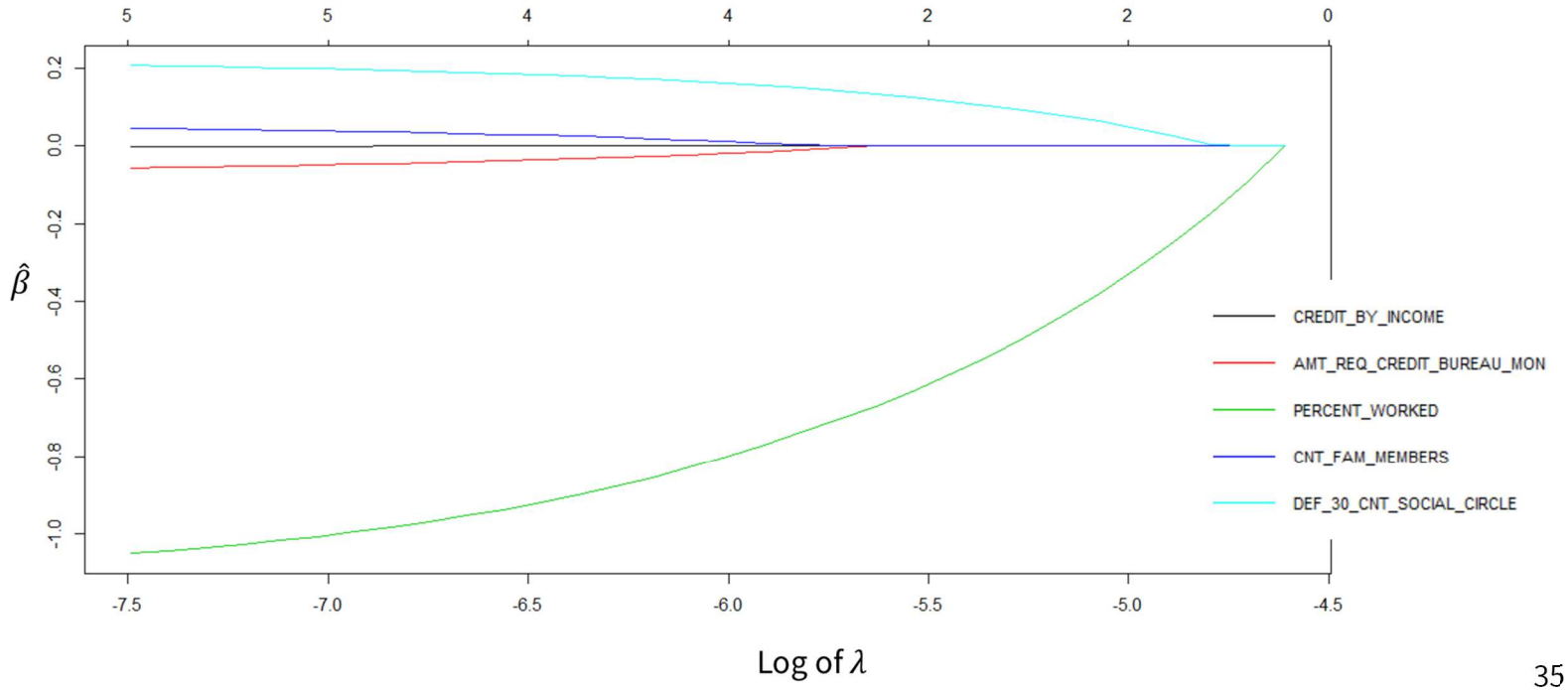
- **Learning algorithm has inbuild mechanism to 'discard' irrelevant features**
- **Often implemented via regularization penalty**
  - LASSO or ridge regression
  - Weight decay in neural networks
  - Extreme gradient boosting
- **Hyperparameter to control regularization**
  - For example λ in ridge regression
  - $J(\boldsymbol{\beta}) = \sum_i (y_i - \boldsymbol{\beta}\boldsymbol{x}_i)^2 + \lambda \sum_j \boldsymbol{\beta}^2$
  - $\boldsymbol{\beta}^* = \min_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$
- **Feature selection?**
  - L2 penalty tends to reduce coefficient values
  - L1 penalty shrinks coefficients to zero



Graphic from Rhys, Hefin I. (2019)

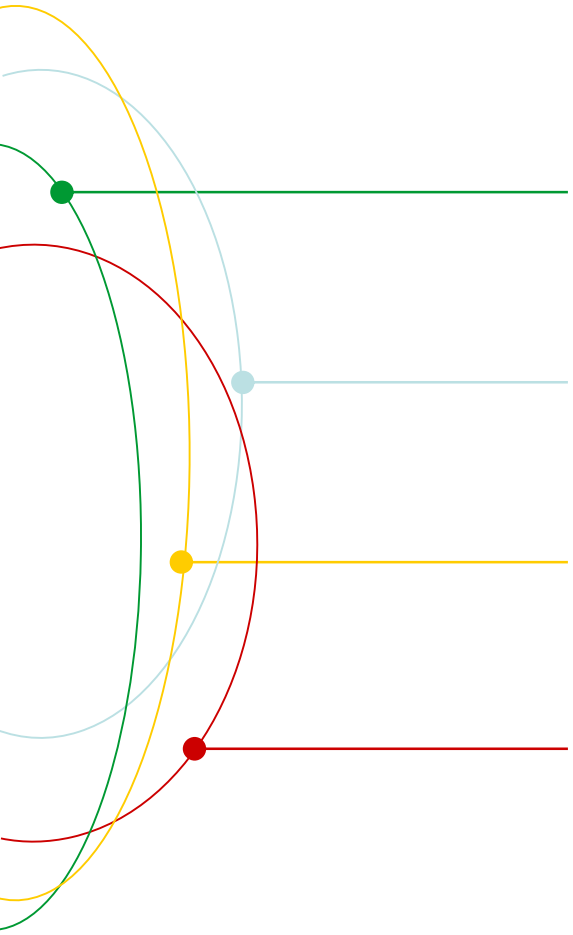
Embedded Methods
Regularization path

- **Estimate regression model with increasing degree of regularization**
- **Compute regularization path w/o re-estimating the model**
  - Friedman et al. (2010) discusses efficient algorithms for several types of models and penalties
- **Example for Home Credit data**



Summary

Summary



Learning goals

- Need and scope of feature engineering (FE)
- Strategies for feature selection

Findings

- FE as a manual activity based on expertise
- Transformations and data-driven FE
- Need for parameter tuning may emerge
- Filters use statistics to judge feature importance
- Wrappers use the prediction model
- Feature ranking versus selection

What next

- Interpretable machine learning
- Diagnose and communicate model results

Literature

- Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software,* 33(1), 1-22.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273-324.
- Kuhn, M., Johnson, K. (2019). Feature Engineering and Selection, CRC Press.
- Niculescu-Mizil, A., Perlich, C., Swirszcz, G., Sindhwani, V., Liu, Y., Melville, P., Wang, D., Xiao, J., Hu, J., Singh, M., Shang, W. X., & Zhu, Y. F. (2009). Winning the KDD Cup Orange Challenge with Ensemble Selection. *Journal of Machine Learning Research: Workshop and Conference Proceedings, 7*, 23-34.
- Rhys, Hefin I. (2019). Machine Learning with R, tidyverse, and mlr. Manning Publications. https://livebook.manning.com/book/machine-learning-for-mortals-mere-and-otherwise/welcome/v-7/

Thank you for your attention!

Stefan Lessmann

Chair of Information Systems
School of Business and Economics
Humboldt-University of Berlin, Germany

Tel. +49.30.2093.5742
Fax. +49.30.2093.5741

stefan.lessmann@hu-berlin.de
http://bit.ly/hu-wi

www.hu-berlin.de