

# Inference and explanation of prediction models



Prof Dr Marko Robnik-Šikonja

Intelligent Systems, November 2020

# Overview of topics

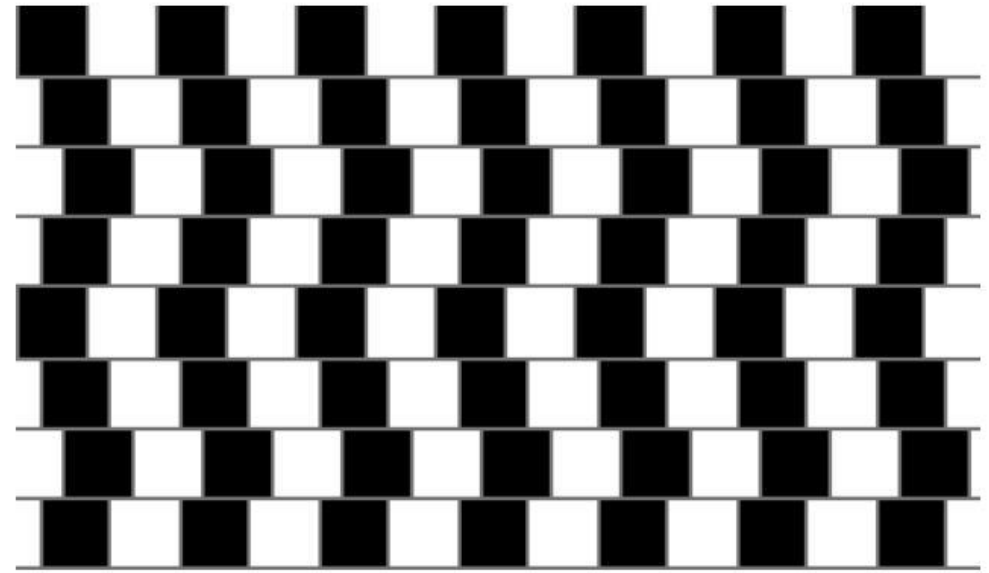
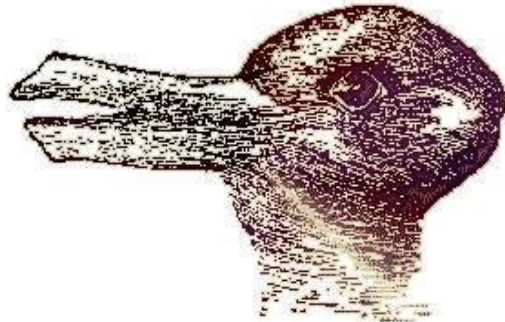
- Model comprehensibility, visualization and knowledge discovery.
- General methodology for explaining predictive models.
- Model level and instance level explanations, methods EXPLAIN and IME.
- Learning with special settings: imbalanced data, cost-sensitive learning
- Calibration of probabilities: binning, isotonic regression.

# Visualization

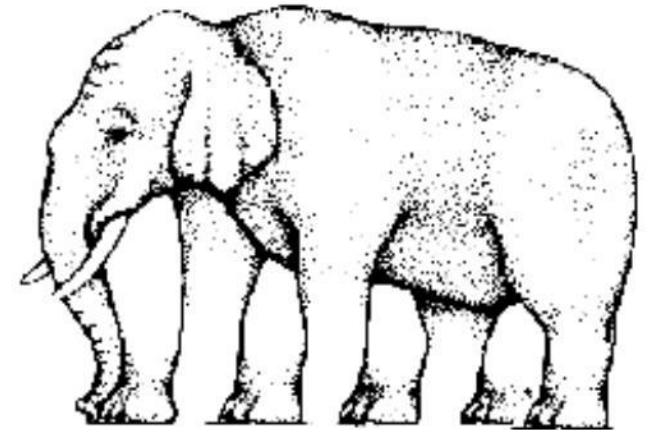
- 1<sup>st</sup> rule of data mining: know your data.
- Therefore: visualizations, getting background data.
- Visualize: distributions of individual variables, their relations, etc.
- For high dimensional data sets one can use scaling.
- Clustering is useful in supervised tasks to get insight into the relation between predicted values  $Y$  and basic groups in the data. If unrelated, feature set might need amendments.

# Visualizations

- Human visual perception has certain limitations:
  - we see what we want to see
  - we see what we see often
  - it is more difficult to notice unexpected patterns
- practice in detection of unknown
- use visualizations which expose “the unknown”



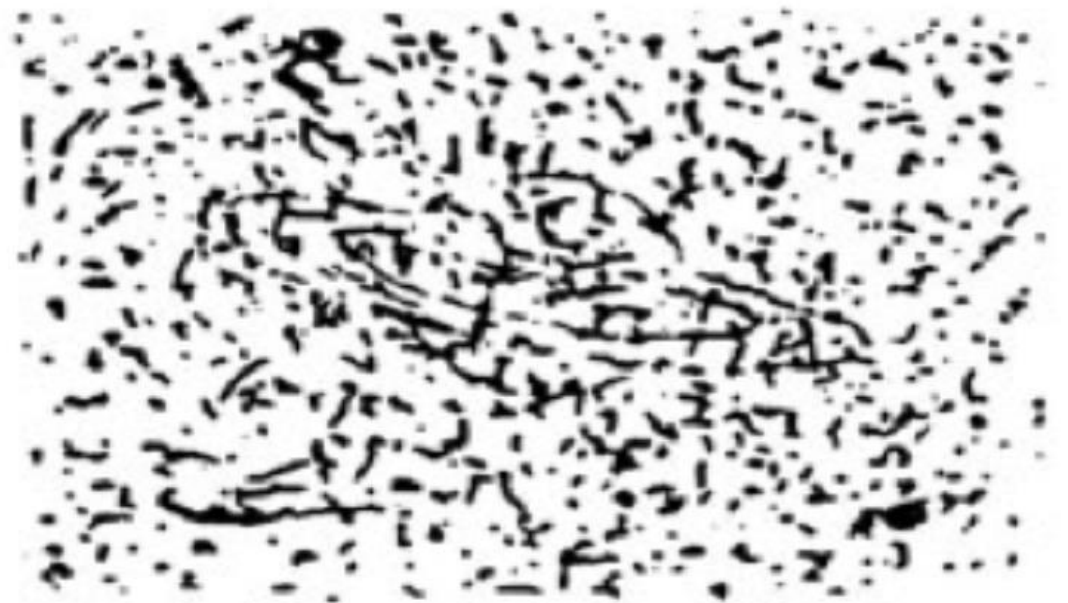
Are the horizontal lines parallel or do they slope?



How many legs does this elephant have?

# Human pattern recognition

- We see inexistent patterns because we WANT to see them (we feel lost without them).



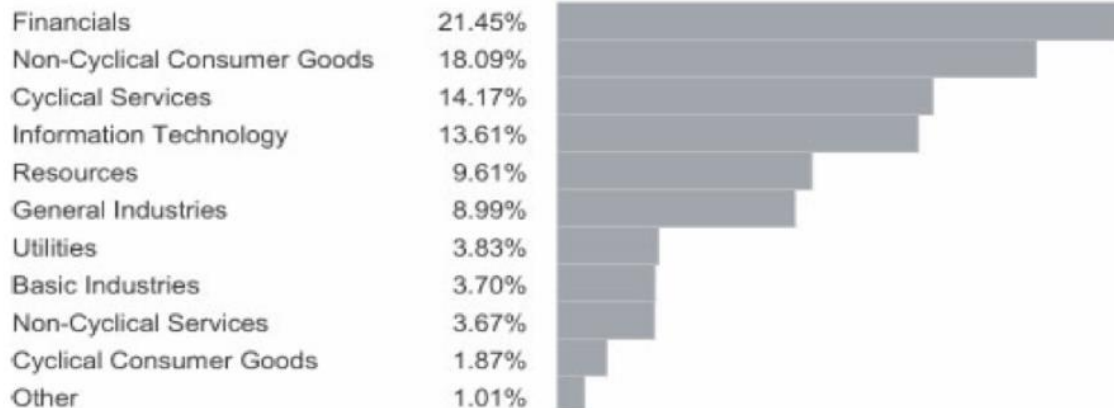
“The researchers found that when people were primed to feel out of control, they were more likely to see patterns where none exist.” (**See a Pattern on Wall Street?**, [John Tierney](#), *Science*)

# Facts about simple visualizations

- Pie charts are a bad choice: hard to read, similar colors, slope, legend is too far away
- Bar chart is much better



## Sector Allocation of Holding

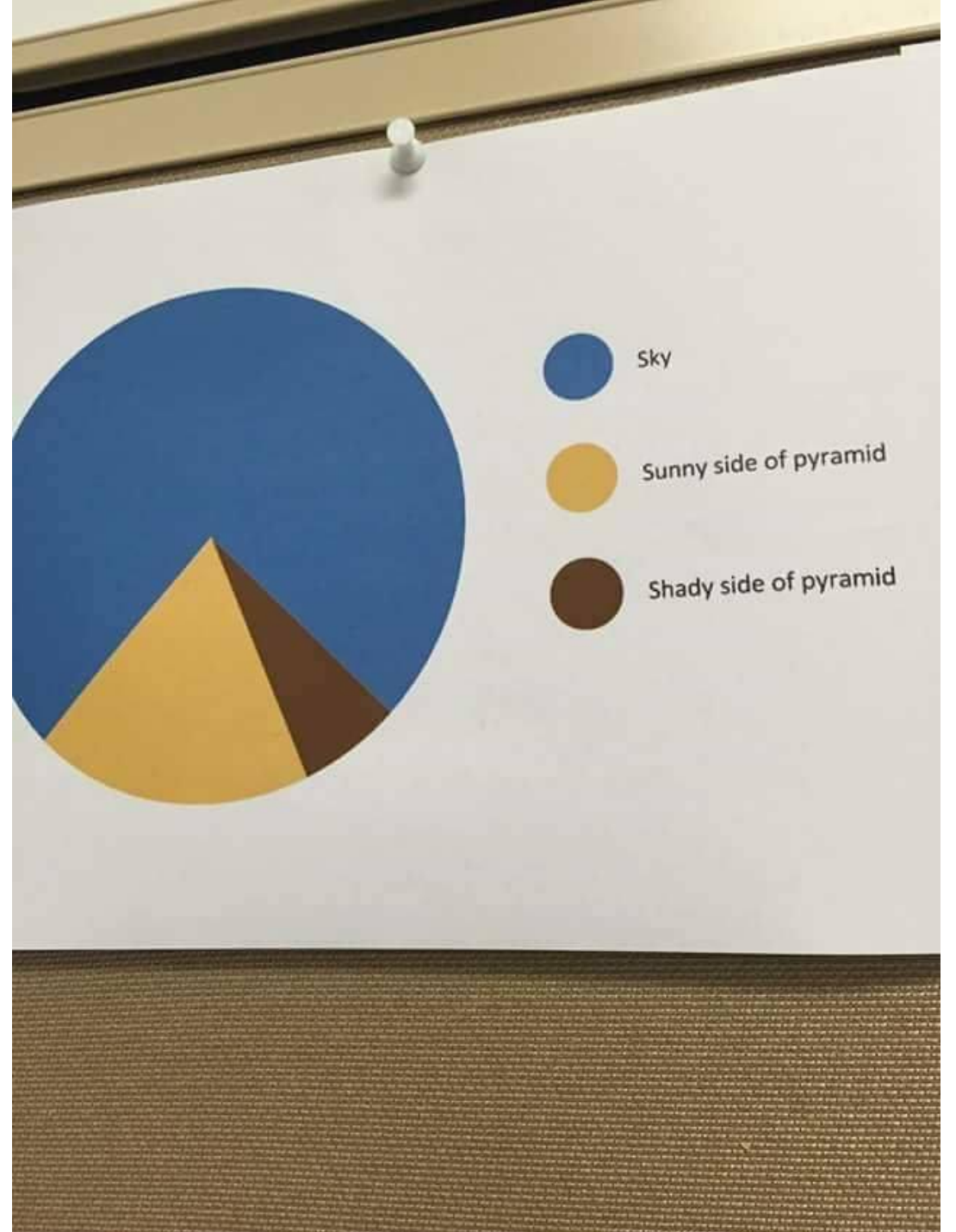


The best pie chart



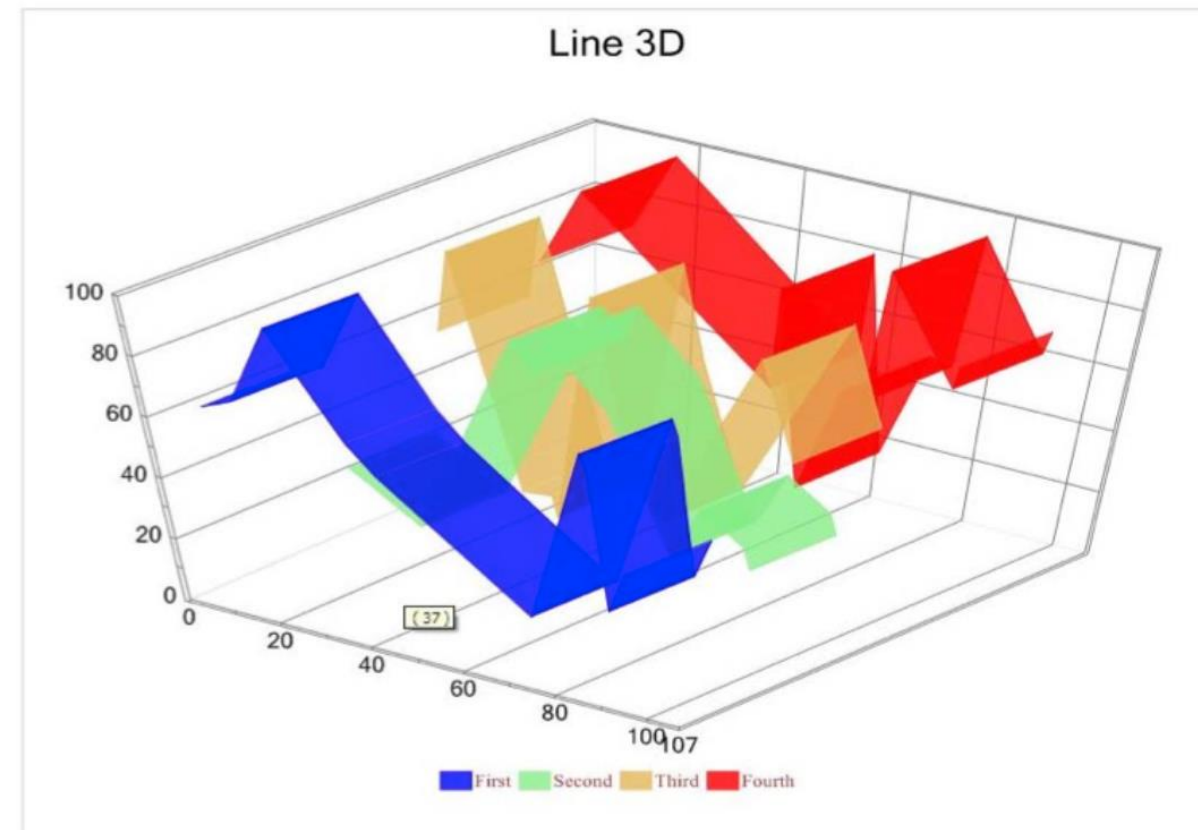
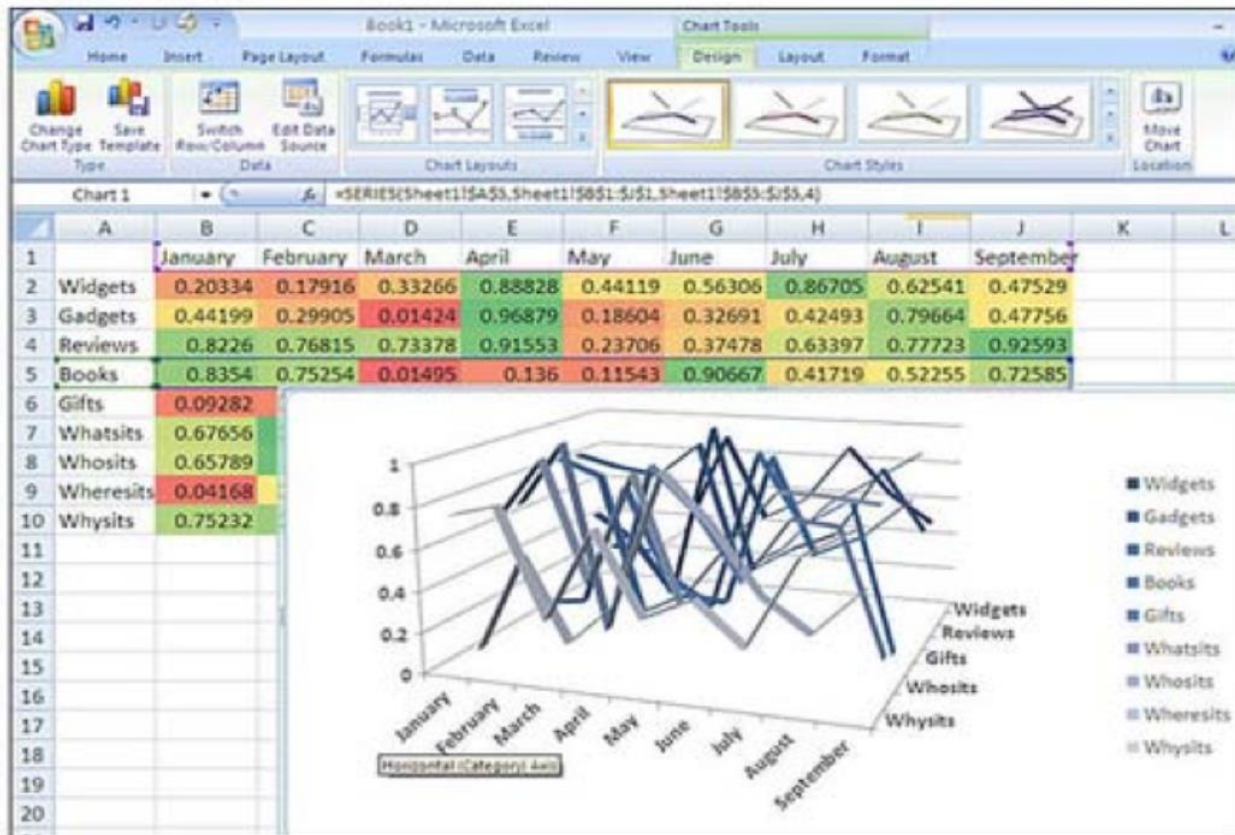
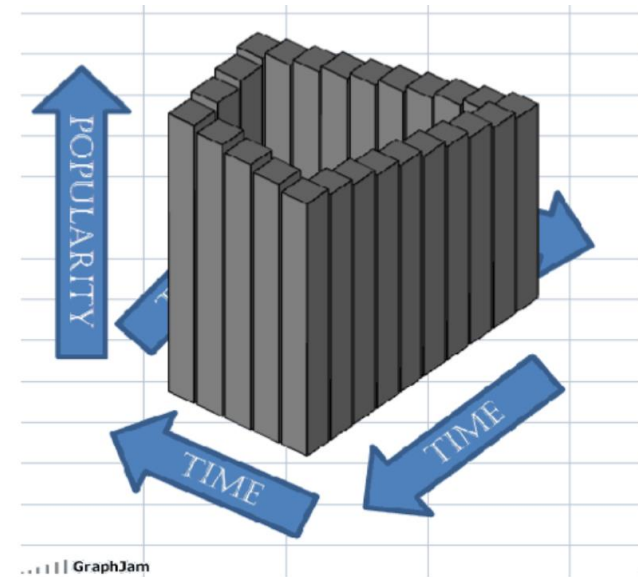
# Pie charts jokes

- notoriously bad



# Facts about simple visualizations

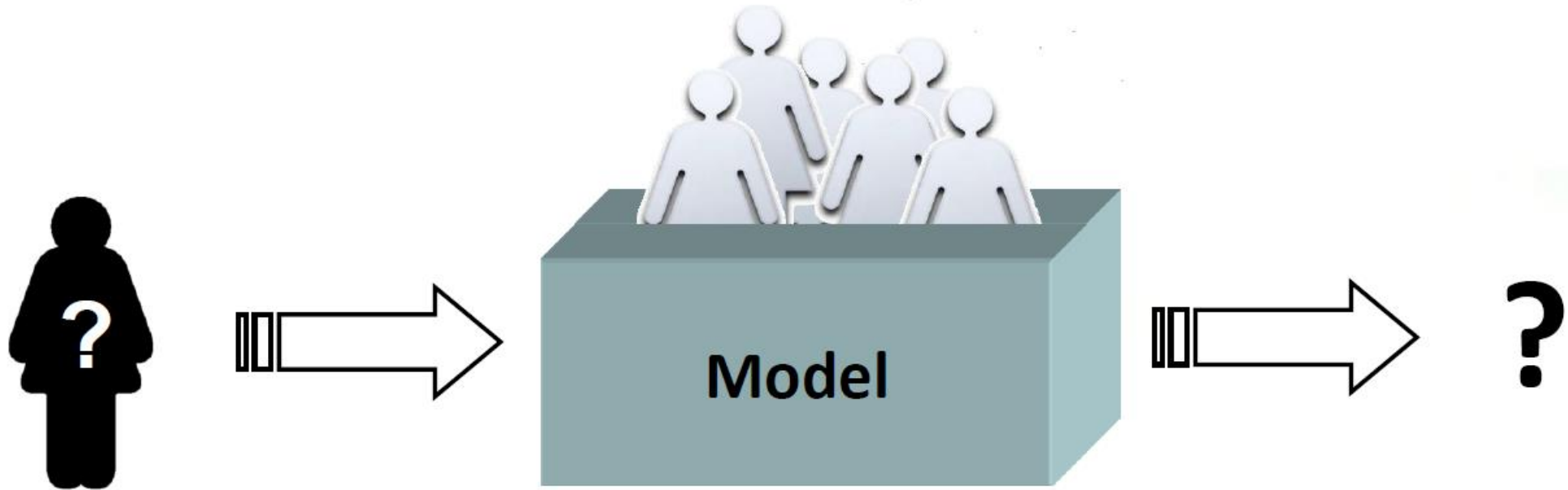
- bar charts, box plots can be OK
- 3D graphs are almost never OK for 2D info: spider plot, bowl of noodles
- take care to be clear and do not manipulate





# Predictive modeling scenario

We want to learn from past examples, with known outcomes.



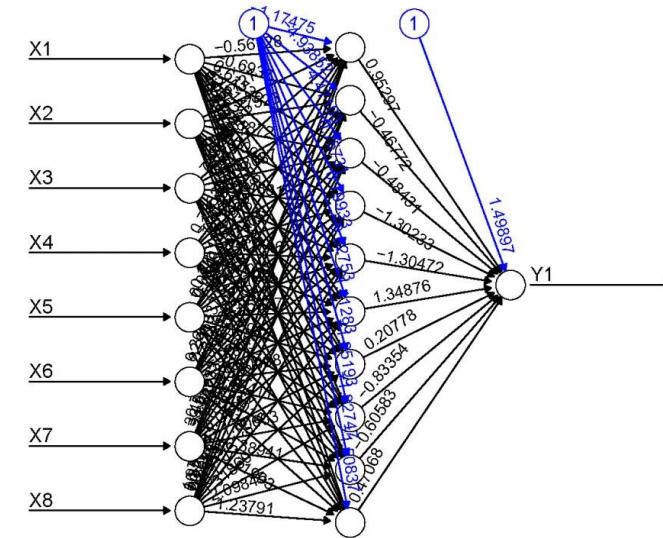
To predict the outcome for a new patient.

# Explanation of predictions

- a number of successful prediction algorithms exist (SVM, boosting, random forests, neural networks), but to a user they are



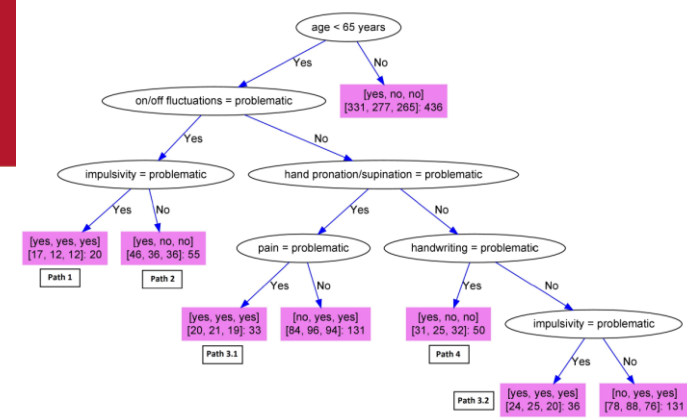
- many fields where users are very much concerned with the transparency of the models: medicine, law, consultancy, public services, etc.
- Goal: a general method applicable to an arbitrary predictor.





# Model comprehensibility

- decision support: model comprehensibility is important to gain users' trust
- knowledge acquisition
- some models are inherently interpretable and comprehensible
- decision and regression trees, classification and regression rules, linear and logistic regression
- really?



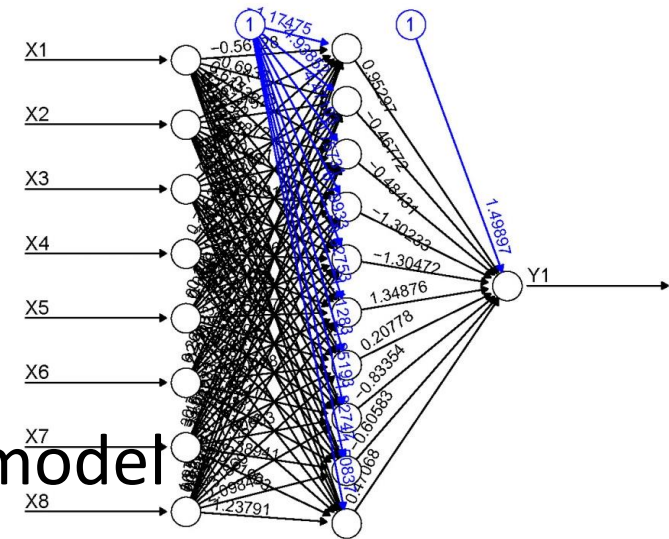
# Domain level explanation

- trying to explain the “true causes and effects”
  - physical processes
  - stock exchange events
- usually unreachable except for artificial problems with known relations and generator function
- some aspects are covered with attribute evaluation, detection of redundancies, ...
- targeted indirectly through the models



# Model-based explanations

- make transparent the prediction process of a particular model
- the correctness of the explanation is independent of the correctness of the prediction but
- better models (with higher prediction accuracy) enable in principle better explanation at the domain level
- we are mostly interested only in the explanation at the model level and leave to the developer of the model the responsibility for its prediction accuracy

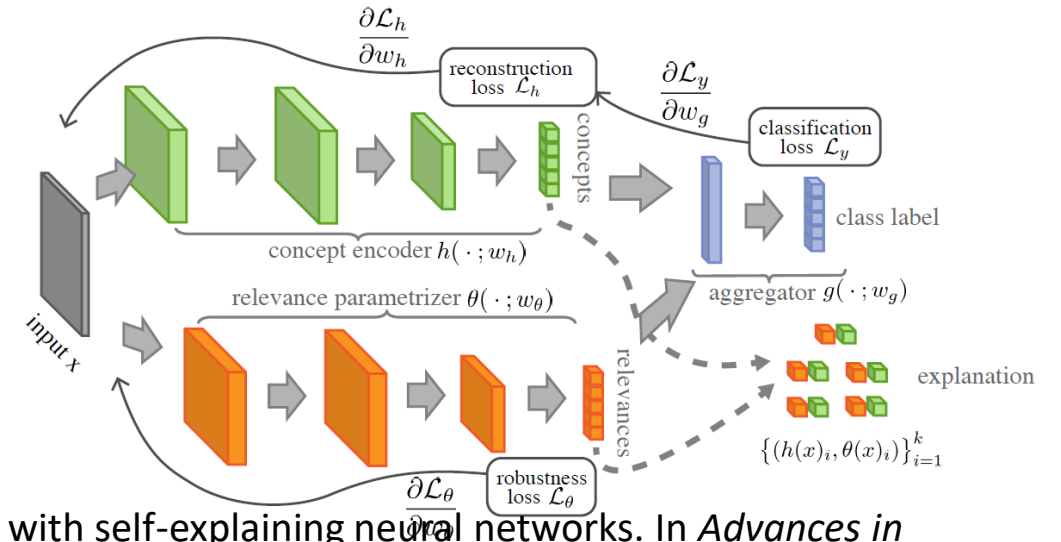


# Two flavours of explanation techniques

- model specific
  - especially used for deep neural networks

Melis, D.A. and Jaakkola, T., 2018. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems* (pp. 7786-7795).

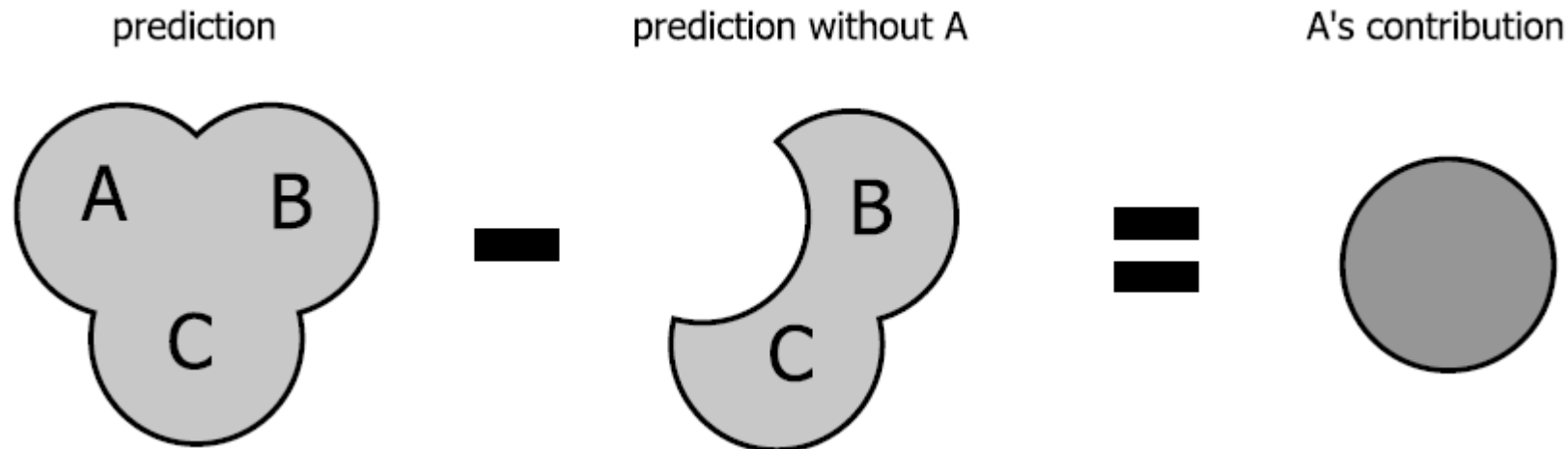
- model agnostic
  - can be used for any predictor,
  - based on perturbation of the inputs





# Idea of perturbation-based explanations

- importance of a feature or a group of features in a specific model can be estimated by simulating lack of knowledge about the values of the feature(s)

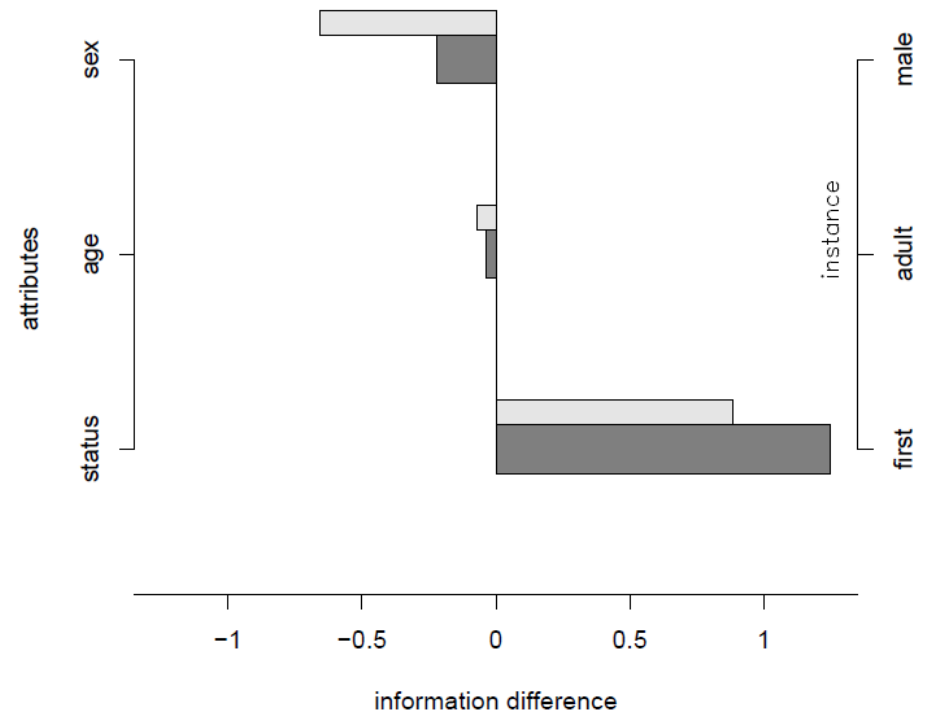




# Instance-level explanation

- explain predictions for each instance separately
  - this is what practitioners applying models are interested in
  - presentation format: impact of each feature on the prediction value
- model-based

Data set: titanic; model: naive Bayes  
 $p(\text{survived}=\text{yes}|\mathbf{x}) = 0.50$ ; true survived=yes

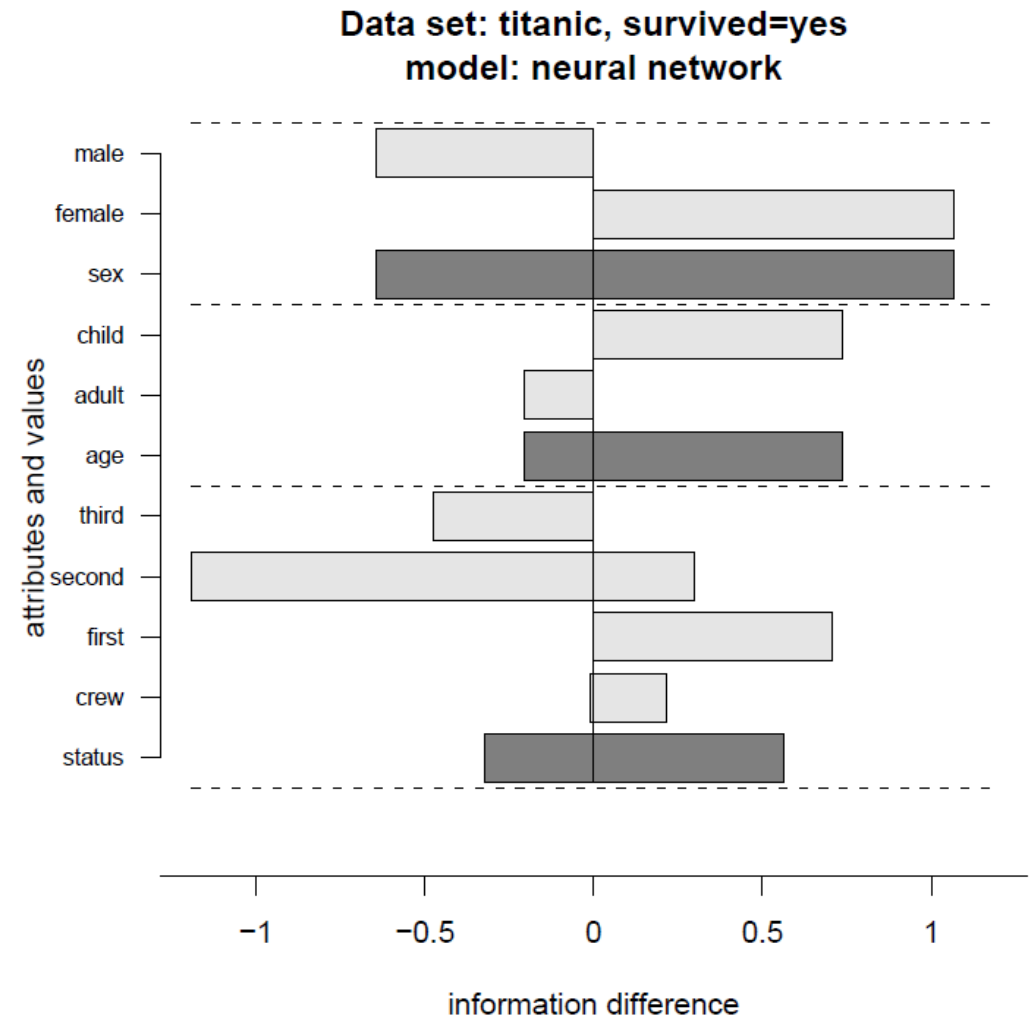






# Model-level explanation

- the overall picture of a problem the model conveys
  - this is what knowledge extractors are interested in
  - presentation format: overall importance of each feature, but also rules, trees
- model-based



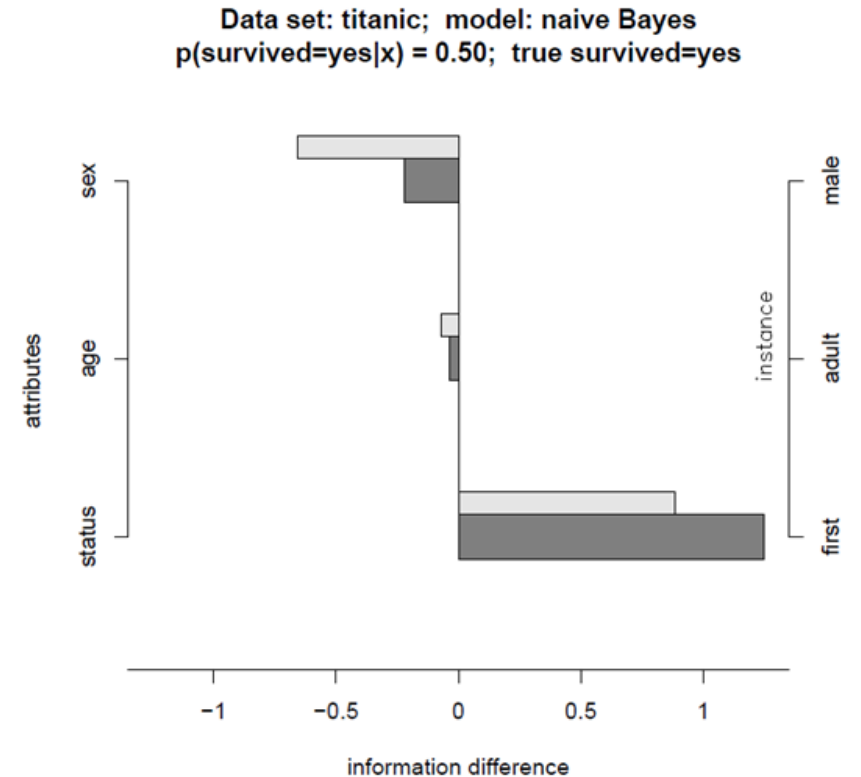


# The method EXPLAIN

- “hide” one attribute at a time
- estimate contribution of attribute from

$$p(y_k | x) - p_{S \setminus \{i\}}(y_k | x)$$

Robnik-Sikonja, M., & Kononenko, I. (2008). Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5), 589-600.



# Explaining EXPLAIN

- assume an instance  $(\mathbf{x}, y)$ , components of  $\mathbf{x}$  are values of attributes  $A_i$
- for a new instance  $\mathbf{x}$ , we want to know what role each attribute's value play in the prediction model  $f$ , i.e. to what extent it contributed to the classification  $f(\mathbf{x})$
- for that purpose
  - we compute  $f(\mathbf{x} \setminus A_i)$ , the model's prediction for  $\mathbf{x}$  without the knowledge of the event  $A_i = a_k$  (marginal prediction)
  - we comparing  $f(\mathbf{x})$  and  $f(\mathbf{x} \setminus A_i)$  to assess importance of  $A_i = a_k$
  - the larger the the difference the more important the role of  $A_i = a_k$  in the model
- $f(\mathbf{x})$  and  $f(\mathbf{x} \setminus A_i)$  are source of explanations

# Evaluation of prediction differences

- how to evaluate  $f(\mathbf{x}) - f(\mathbf{x} \setminus A_i)$
- in classification we take  $f(\mathbf{x})$  in the form of probability

## 1. difference of probabilities

$$\text{probDiff}_i(y|\mathbf{x}) = p(y|\mathbf{x}) - p(y|\mathbf{x} \setminus A_i)$$

## 2. information gain (Shannon, 1948)

$$\text{infGain}_i(y|\mathbf{x}) = \log_2 p(y|\mathbf{x}) - \log_2 p(y|\mathbf{x} \setminus A_i)$$

## 3. weight of evidence also log odds ratio (Good, 1950)

$$\text{odds}(z) = p(z) / (1 - p(z))$$

$$\text{WE}_i(y|\mathbf{x}) = \log_2 \text{odds}(y|\mathbf{x}) - \log_2 \text{odds}(y|\mathbf{x} \setminus A_i)$$

# Implementation

- $p(y | \mathbf{x})$ : classify  $\mathbf{x}$  with the model
- $p(y | \mathbf{x} \setminus A_i)$  – simulate lack of knowledge of  $A_i$  in the model
  - replace with special NA value: good for some, mostly bad, left to the mercy of model's internal mechanism
  - average prediction across perturbations of  $A_i$   
$$p(y | \mathbf{x} \setminus A_i) = \sum_a p(A_i = a_s) p(y | \mathbf{x} \leftarrow A_i = a_s)$$
  - use discretization for numeric attributes
  - use Laplace correction for probability estimation



# Weaknes of EXPLAIN

- “hide” one attribute at a time
- estimate contribution of attribute from

$$p(y_k | x) - p_{S \setminus \{i\}}(y_k | x)$$

- weakness: if there are redundant ways to express concept, credit is not assigned
- example:

$$C = A_1 \vee A_2 A_3$$

explanation for instance ( $A_1=A_2=A_3=1$ )

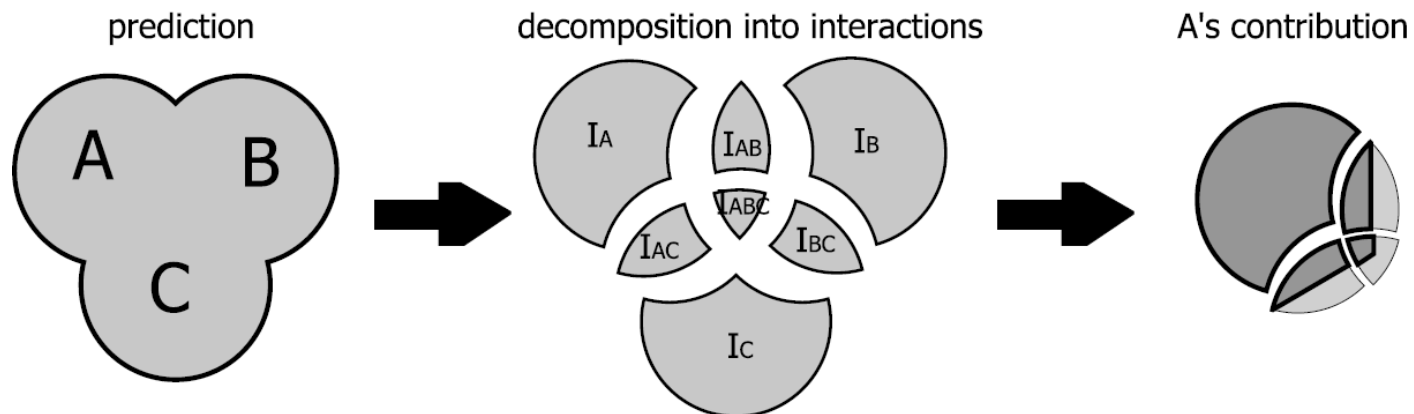


# The method IME

- (Interactions-based Method for Explanation)
- “hide” any subset of attributes at a time ( $2^a$  subsets!)
- the source of explanations is the difference in prediction using a subset of features  $Q$  and an empty set of features  $\{\}$

$$\Delta_Q = h(x_Q) - h(x_{\{\}})$$

- the feature gets some credit for standalone contributions and for contributions in interactions





# IME: sum over all subsets

- the contributions are

$$\pi_i = \sum_{Q \subseteq \{1, 2, \dots, a\} - \{i\}} \frac{1}{a \binom{a-1}{a-|Q|-1}} (\Delta_{Q \cup \{i\}} - \Delta_Q)$$

Štrumbelj, E., Kononenko, I. & Robnik-Šikonja, M., Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, Oct. 2009, 68(10):886-904







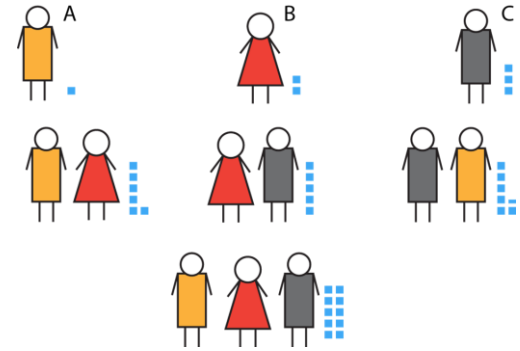
# Game theory analogy

- coalitional game of  $a$  players (attributes)
- players form coalitions (i.e. interactions)
- how to distribute the payout to the members of a coalition: (how to assign the credit for prediction)
- The Shapley value is the unique payoff vector that is
  - efficient (exactly splits payoff value),
  - symmetric (equal payments to equivalent players)
  - additive (overall credit is a sum of participating in coalitions), and
  - assigns zero payoffs to dummy players (no contribution to any coalition).





# Shapley value



$$Sh_i(v) = \sum_{S \subseteq N \setminus \{i\}, s=|S|} \frac{(n-s-1)!s!}{n!} (v(S \cup \{i\}) - v(S)), \quad i = 1, \dots, n.$$

$$\pi_i = \sum_{Q \subseteq \{1, 2, \dots, a\} - \{i\}} \frac{1}{a \binom{a-1}{a-|Q|-1}} (\Delta_{Q \cup \{i\}} - \Delta_Q)$$

- Shapley value can be efficiently approximated





# Solution for IME: sampling

- Shapley value can be expressed in an alternative formulation
- $\pi(a)$  is the set of all ordered permutations of  $a$
- $\text{Pre}^i(\mathcal{O})$  is the set of players which are predecessors of player  $i$  in the order  $\mathcal{O} \in \pi(a)$

$$\begin{aligned}\varphi_i(k, x) &= \frac{1}{a!} \sum_{\mathcal{O} \in \pi(a)} (\Delta(\text{Pre}^i(\mathcal{O}) \cup \{i\})(k, x) - \Delta(\text{Pre}^i(\mathcal{O}))(k, x)) = \\ &= \frac{1}{a!} \sum_{\mathcal{O} \in \pi(a)} (p_{\text{Pre}^i(\mathcal{O}) \cup \{i\}}(y_k | x) - p_{\text{Pre}^i(\mathcal{O})}(y_k | x)),\end{aligned}$$

- smart sampling over subsets of attributes
- computationally feasible approach





# IME algorithm

---

**Algorithm 1** Approximating the contribution of the  $i$ -th feature's value,  $\varphi_i$ , for instance  $x \in \mathcal{A}$ .

---

determine  $m$ , the desired number of samples

$\varphi_i \leftarrow 0$

**for**  $j = 1$  to  $m$  **do**

    choose a random permutation of features  $O \in \pi(N)$

    choose a random instance  $y \in \mathcal{A}$

$v_1 \leftarrow f(\tau(x, y, Pre^i(O) \cup \{i\}))$

$v_2 \leftarrow f(\tau(x, y, Pre^i(O)))$

$\varphi_i \leftarrow \varphi_i + (v_1 - v_2)$

**end for**

$\varphi_i \leftarrow \frac{\varphi_i}{m}$

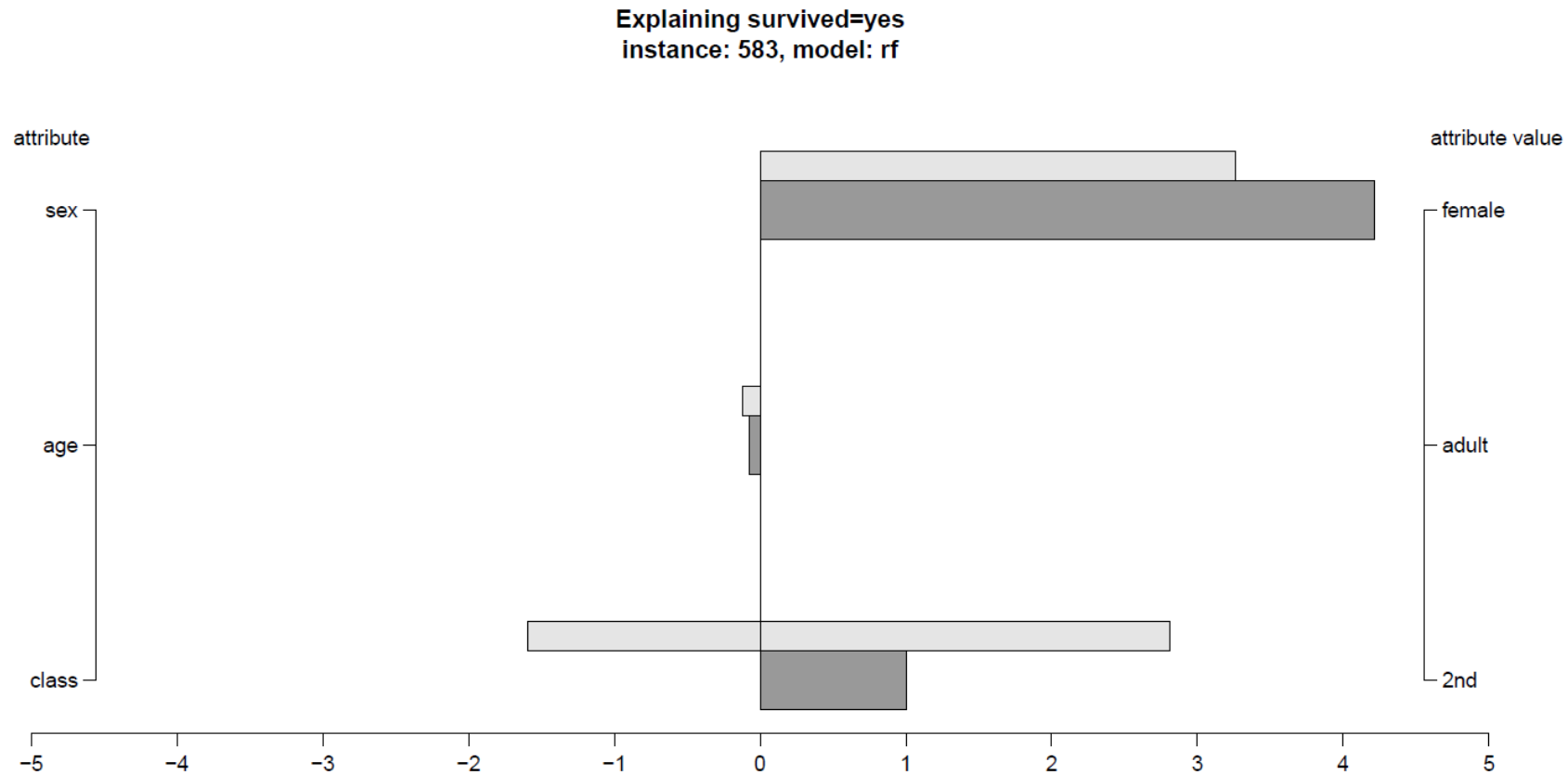
- by measuring the variance of contributions we can determine the necessary number of samples for each attribute





# Visualization of explanations

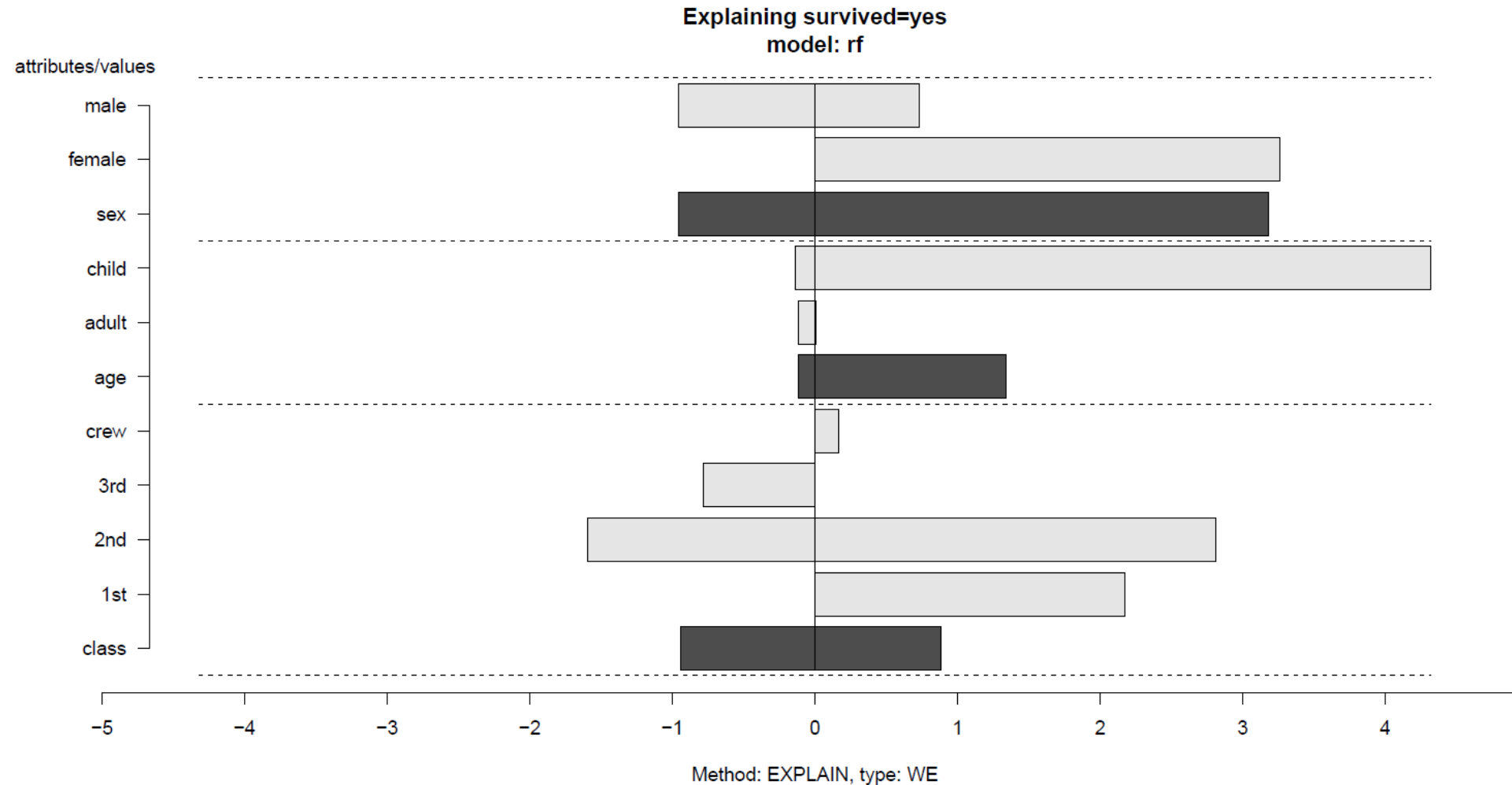
- instance-level explanation on Titanic data set





# Visualization of explanations

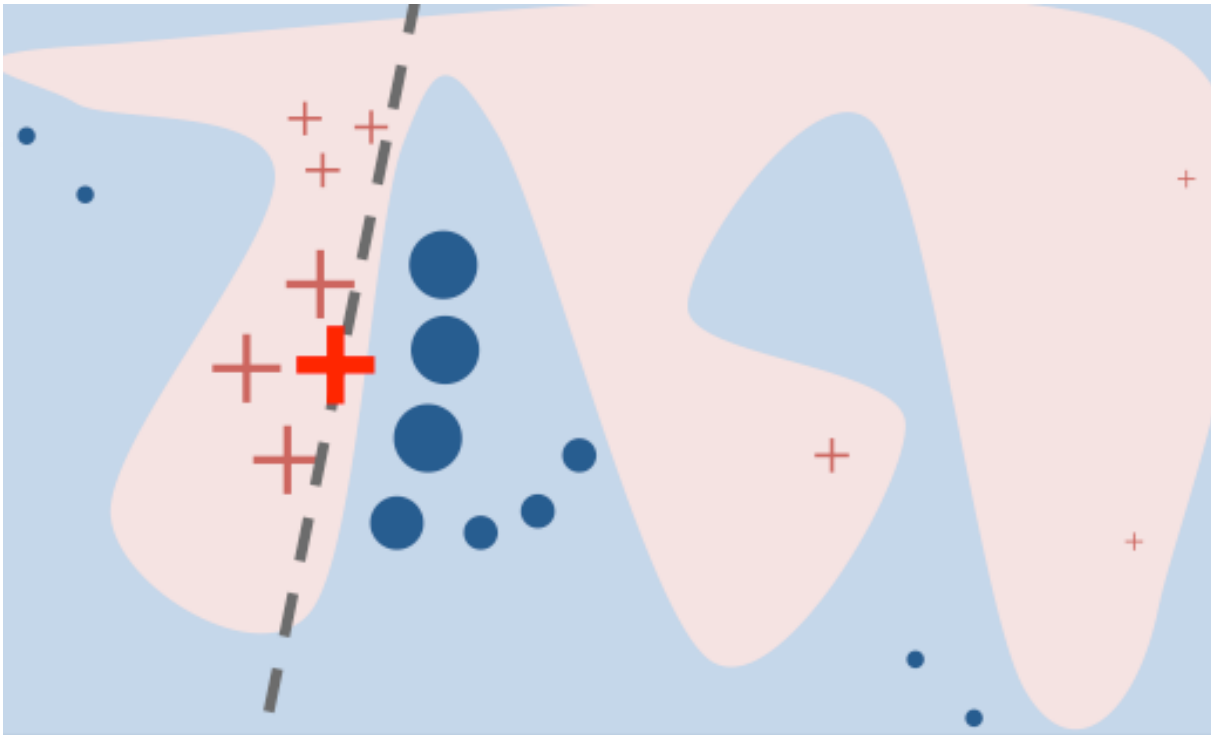
- model-level explanation on Titanic data set





# LIME explanation method

- Local Interpretable Model-agnostic Explanations)
- perturbations in the locality of an explained instance

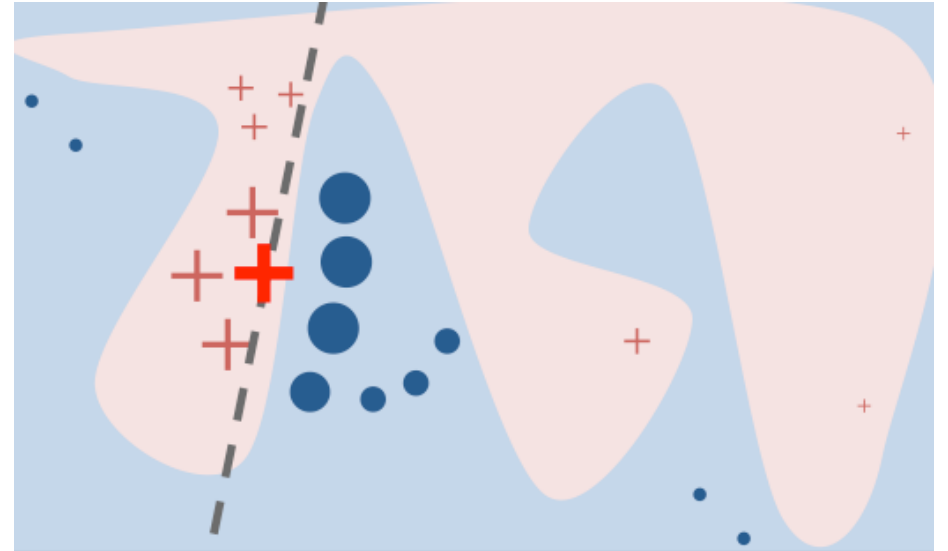




# LIME explanation method

- optimize a trade-off between local fidelity of explanation and its interpretability

$$e(x) = \arg \min_{g \in G} L(f, g, \pi) + \Omega(g)$$



- L is a local fidelity function, f is a model to be explained, g is an interpretable local model g (i.e. linear model),  $\pi(x, z)$  is proximity measure between the explained instance x and perturbed points z in its neighborhood,  $\Omega$  is a model complexity measure

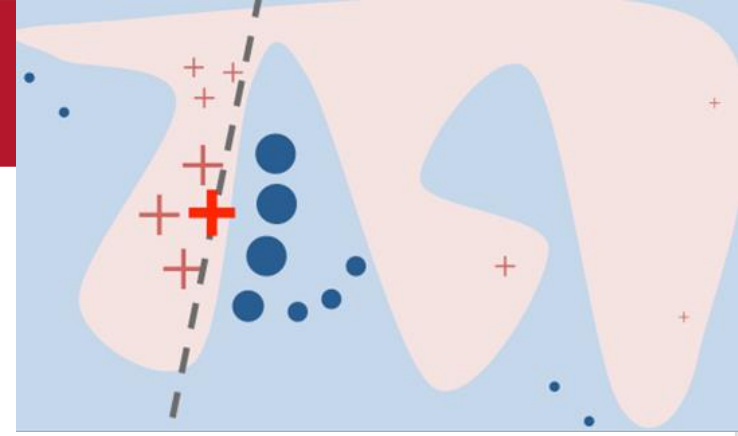






# LIME details

- samples around explanation instance  $x$  to draw samples  $z$  weighted by the distance  $\pi(x, z)$
- samples  $z$  are used to training an interpretable model  $g$  (linear model)
- the squared loss measures local infidelity
- number of non-zero weights is complexity
- samples are weighted according to the Gaussian distribution of the distance between  $x$  and  $z$





# LIME strengths and weaknesses

- faster than IME
- works for many features, including text and images
- no guarantees that the explanations are faithful and stable
- neighborhood based: a curse of dimensionality
- may not detect interactions due to (too) simple interpretable local model (linear model)





















# SHAP

- SHapley Additive exPlanation
- unification of several explanation methods, including IME and LIME
- KernelSHAP: based on Shapley values which are estimated using a LIME style linear regression
- faster than IME but
- still uses linear model with all its strengths and weaknesses

(A)

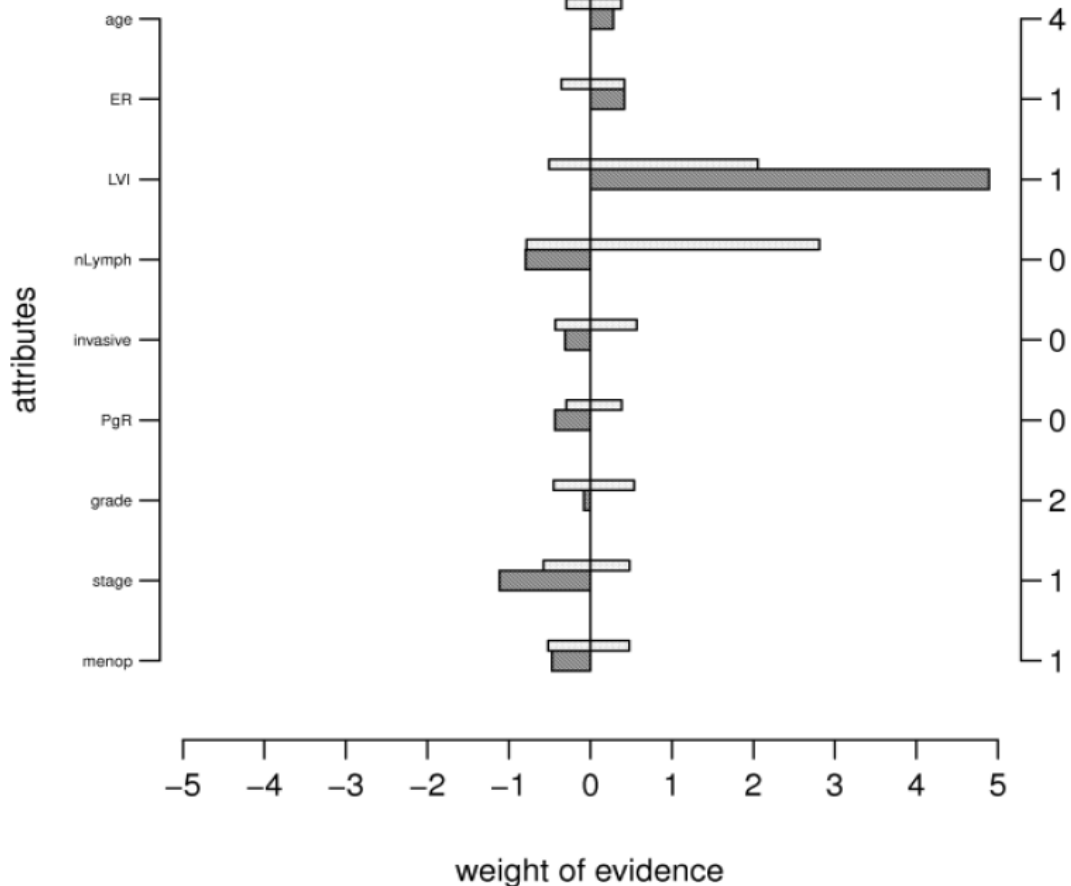
	Input	Explain 8	Explain 3	Masked
Orig. DeepLift				
New DeepLift				
SHAP				
LIME				





# Use case: breast cancer recurrence

Data set: onko; model: PRBF  
 $p(\text{recurrence}=1|x) = 0.81$ ; true recurrence=2



## Cancer recurrence within 10 years

*menop* binary feature indicating menopausal status

*stage* tumor stage 1: less than 20mm, 2: between 20mm and 50mm, 3: over 50mm

*grade* tumor grade 1: good, 2: medium, 3: poor, 4: not applicable, 9: not determined

*histType* histological type of the tumor 1: ductal, 2: lobular, 3: other

*PgR* level of progesterone receptors in tumor (in fmol per mg of protein) 0: less than 10, 1: more than 10, 9: unknown

*invasive* invasiveness of the tumor 0: no, 1: invades the skin, 2: the mamilla, 3: skin and mamilla, 4: wall or muscle

*nLymph* number of involved lymph nodes 0: 0, 1: between 1 and 3, 2: between 4 and 9, 3: 10 or more

*famHist* medical history 0: no cancer, 1: 1st generation breast, ovarian or prostate cancer, 2: 2nd generation breast, ovarian or prostate cancer, 3: unknown gynecological cancer, 4: colon or pancreas cancer, 5: other or unknown cancers, 9: not determined

*LVI* binary feature indicating lymphatic or vascular invasion

*ER* level of estrogen receptors in tumor (in fmol per mg of protein) 1: less than 5, 2: 5 to 10, 3: 10 to 30, 4: more than 30, 9: not determined

*maxNode* diameter of the largest removed lymph node 1: less than 15mm, 2: between 15 and 20mm, 3: more than 20mm

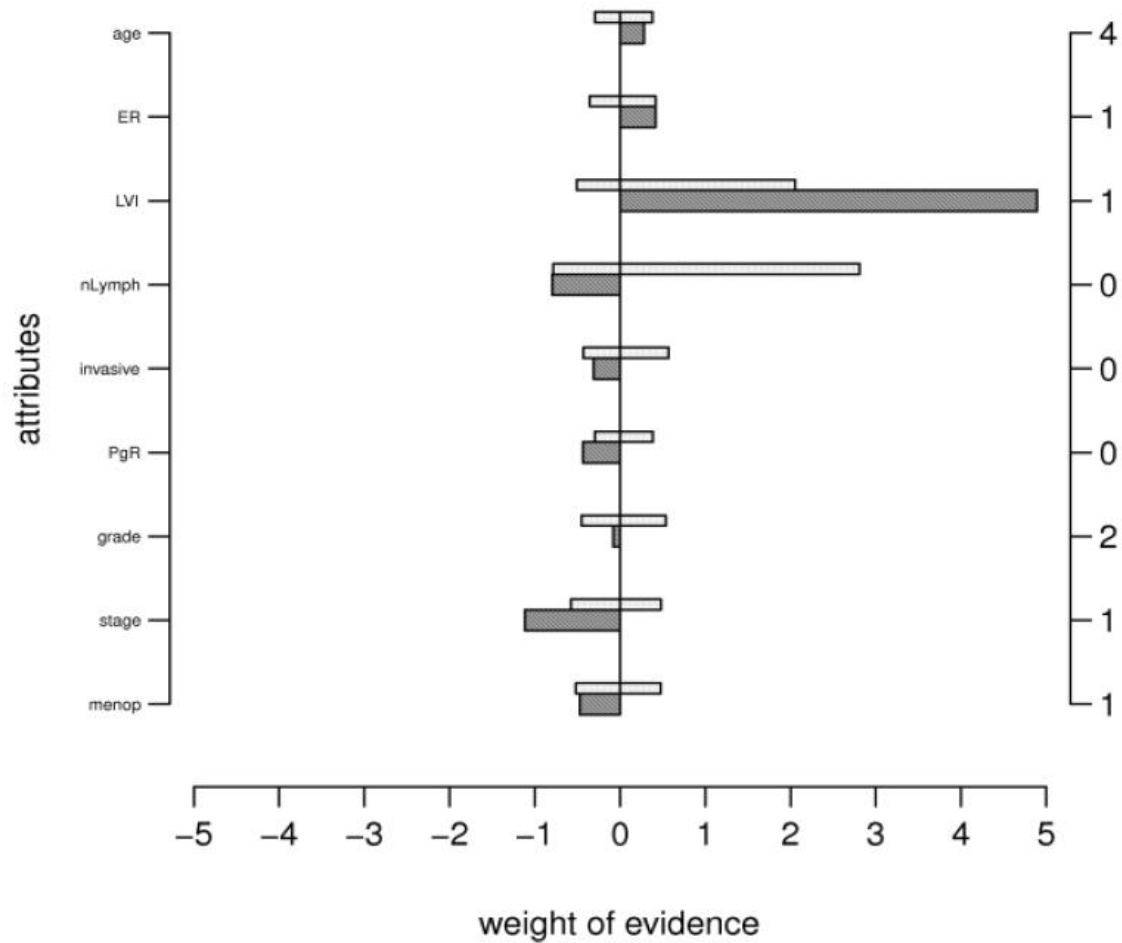
*posRatio* ratio between involved and total lymph nodes removed 1: 0, 2: less than 10%, 3: between 10% and 30%, 4: over 30%

*age* patient age group 1: under 40, 2: 40-50, 3: 50-60, 4: 60-70, 5: over 70 years

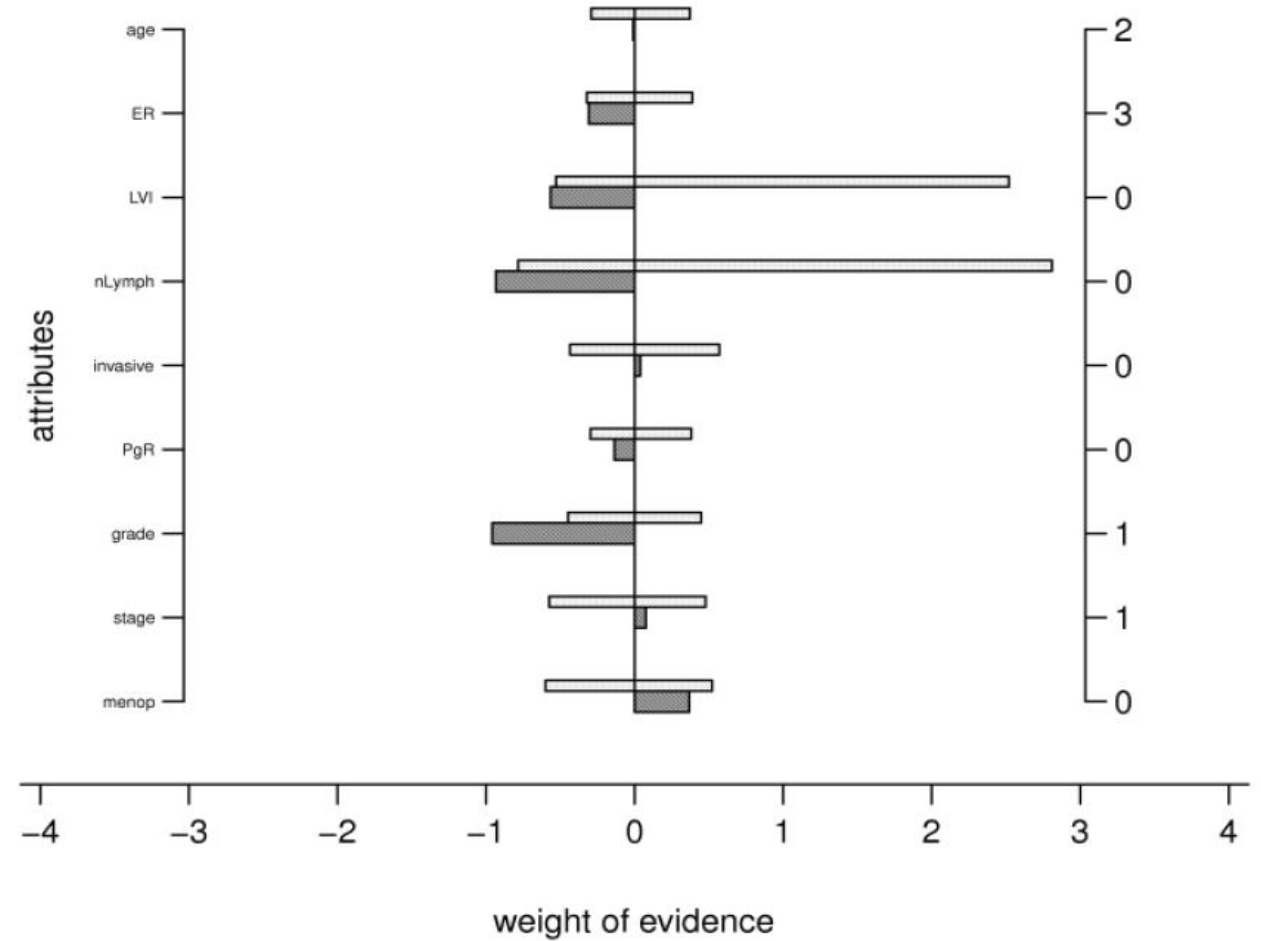


# Use case: breast cancer recurrence

Data set: onko; model: PRBF  
 $p(\text{recurrence}=1|x) = 0.81$ ; true recurrence=2

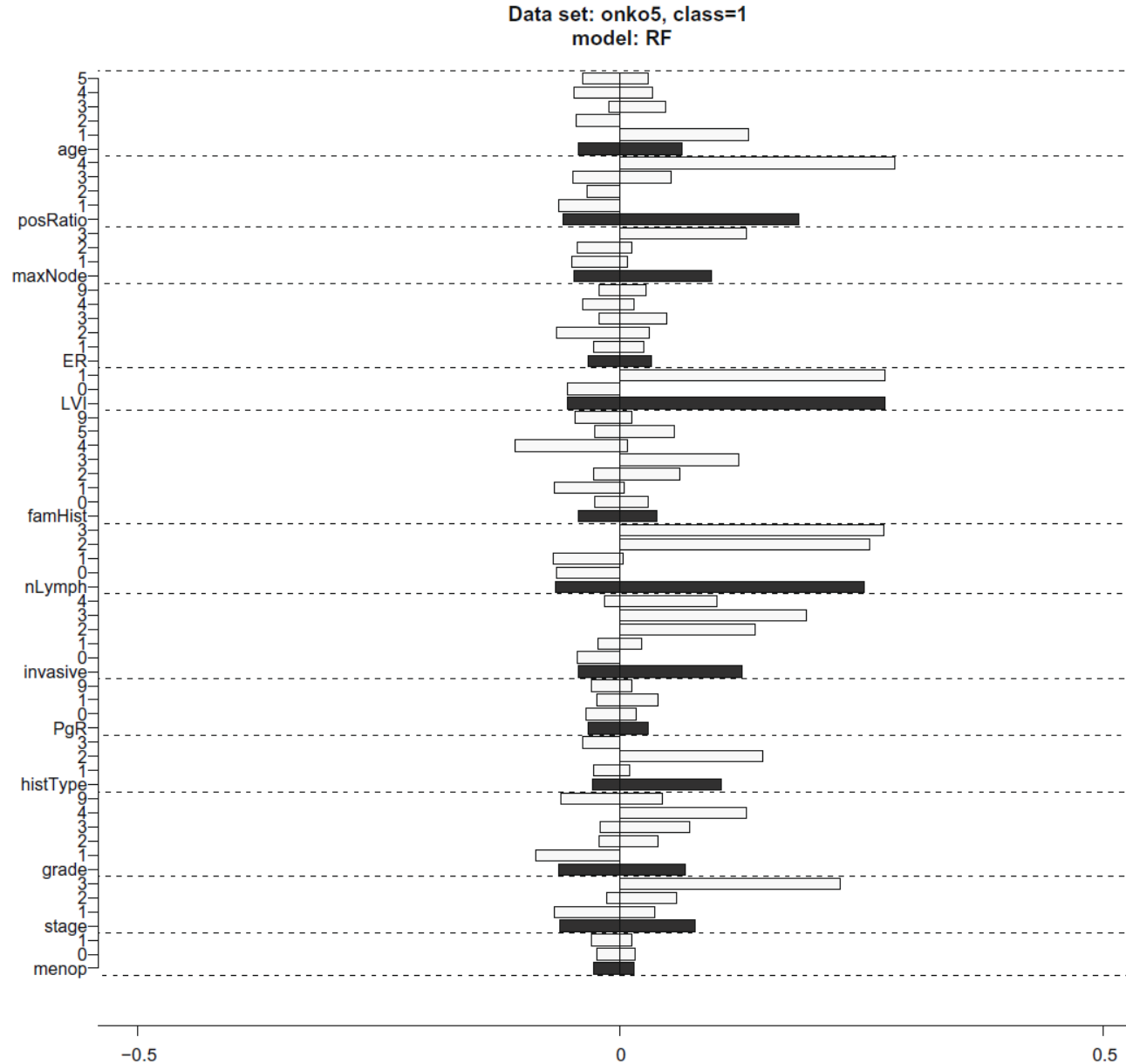


Data set: onko; model: PRBF  
 $p(\text{recurrence}=1|x) = 0.06$ ; true recurrence=2



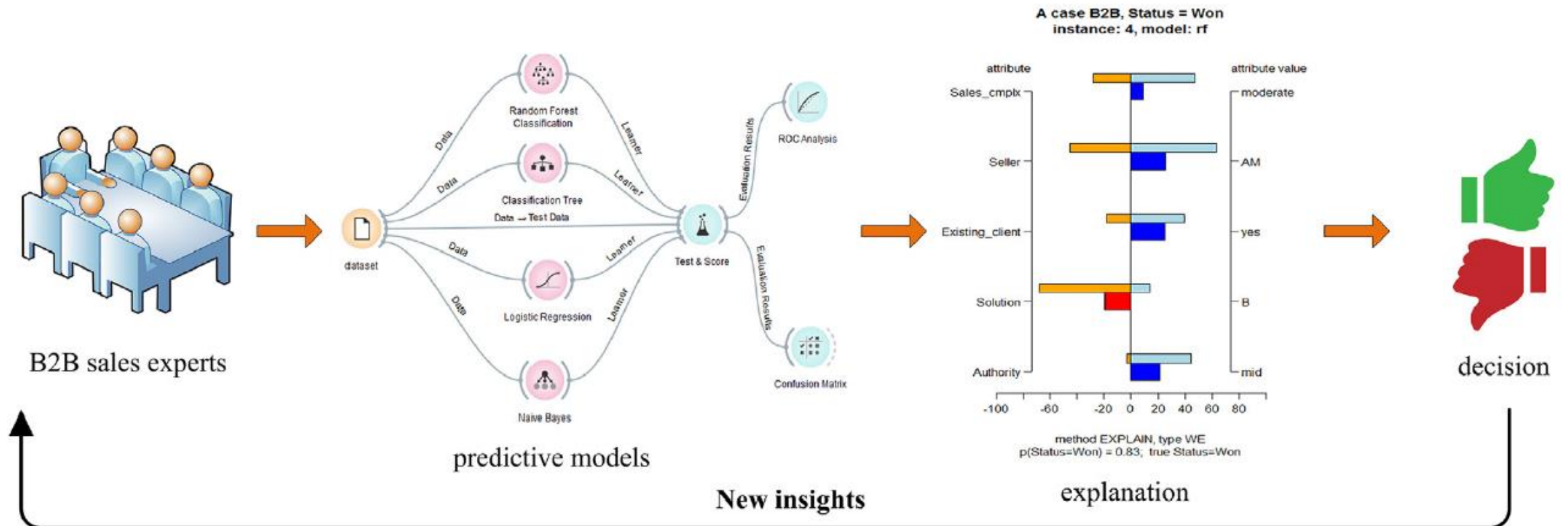


# Use case: breast cancer recurrence



# Use case: B2B sales forecasting

- Goals: improve understanding of factors influencing the outcome and improve the sales performance





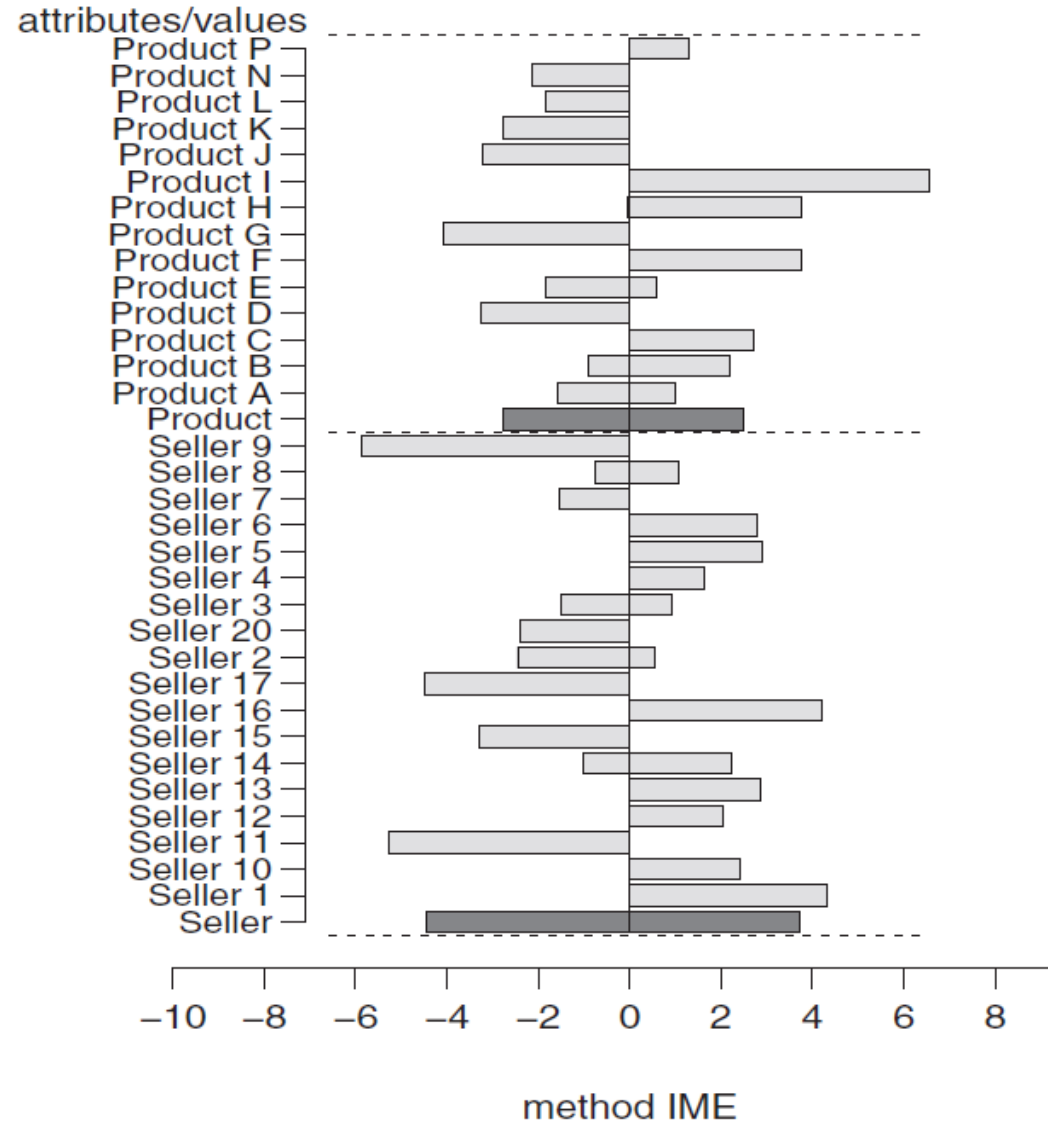
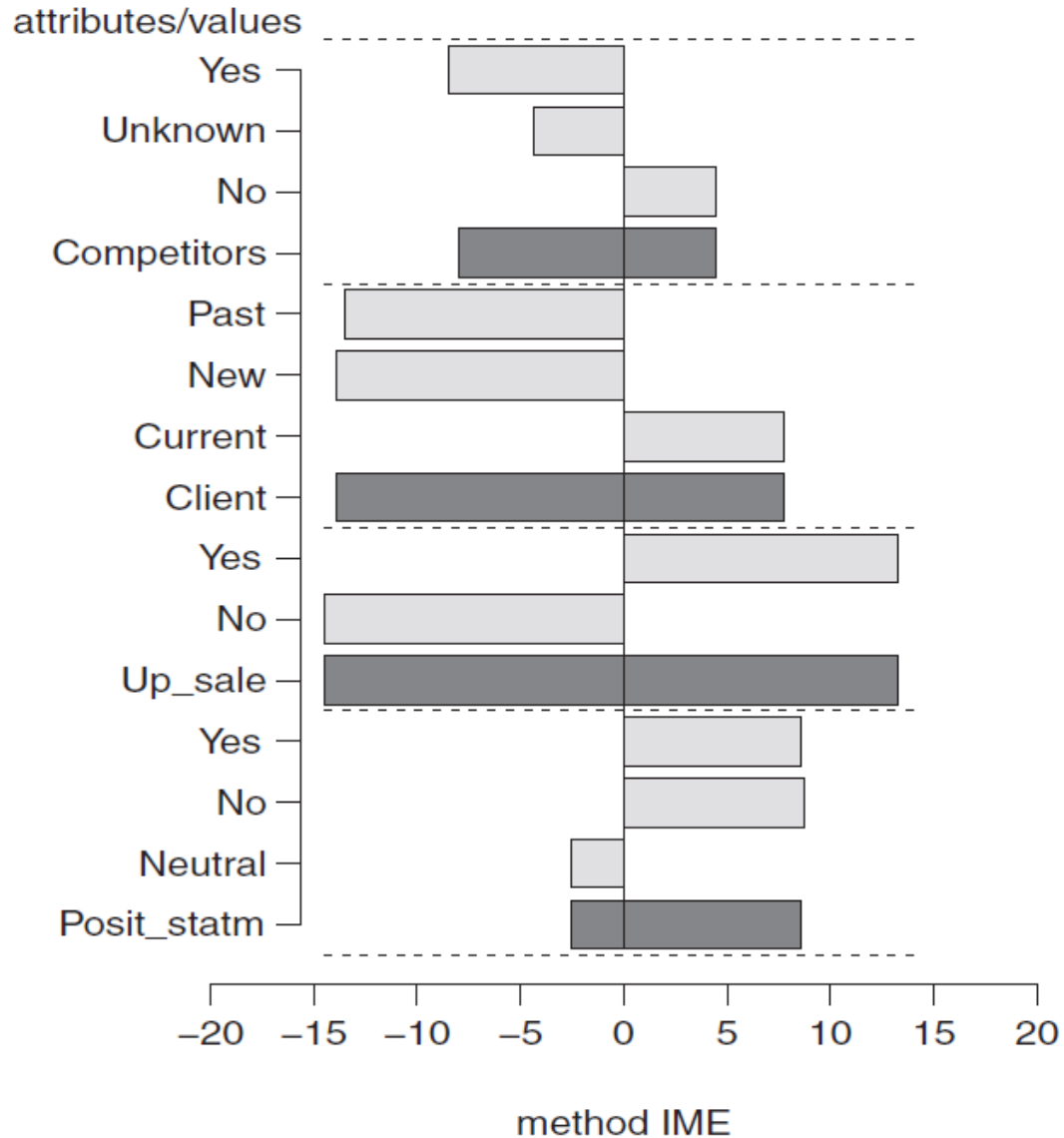
# B2B sales attributes

Attribute	Description	Values
Authority	Authority level at a client side	Low, mid, high
Product	Offered product	e.g. A, B, C, etc.
Seller	Seller's name	Seller's name
Competitors	Do we have competitors?	No, yes, unknown
Company size	Size of a company	Big, mid, small
Purchasing department	Is the purchasing department involved?	No, yes, unknown
Partnership	Selling in partnership?	No, yes
Budget allocated	Did the client reserve the budget?	No, yes, unknown
Formal tender	Is a tendering procedure required?	No, yes
RFI	Did we get request for information?	No, yes
RFP	Did we get request for proposal?	No, yes
Growth	Growth of a client?	Growth, stable, etc.
Positive statements	Positive attitude expressed?	No, yes, neutral
Source	Source of the opportunity	e.g. referral, web, etc.
Client	Type of a client	New, current, past
Cross sale	A different product to existing client?	No, yes
Scope clarity	Implementation scope defined?	Clear, few questions, etc.
Strategic deal	Does this deal have a strategic value?	Very important, etc.
Up sale	Increasing sales of existing products?	No, yes
Deal type	Type of a sale	Consulting, project, etc.
Needs defined	Is client clear in expressing the needs?	Info gathering, etc.
Attention to client	Attention to a client	First deal, normal, etc.
Status	An outcome of sales opportunity	Lost, won



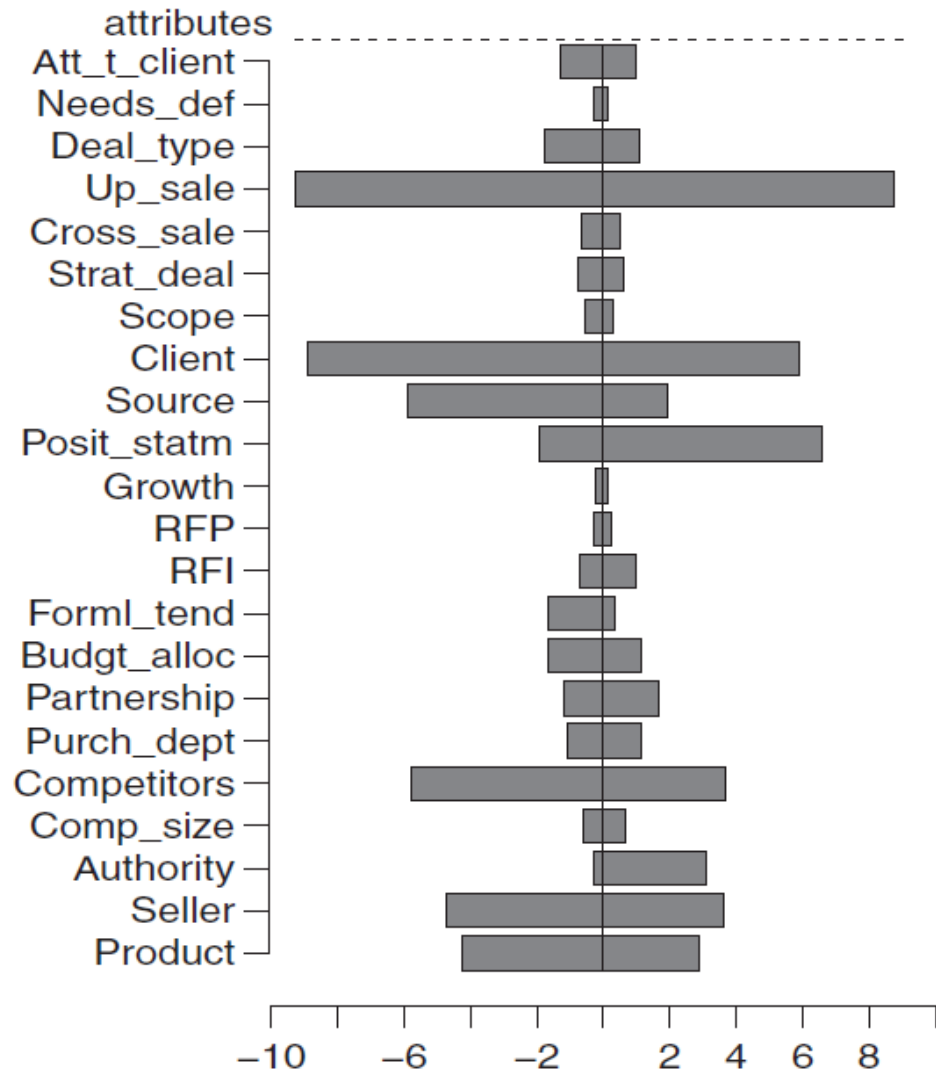


# B2B sales: drill in

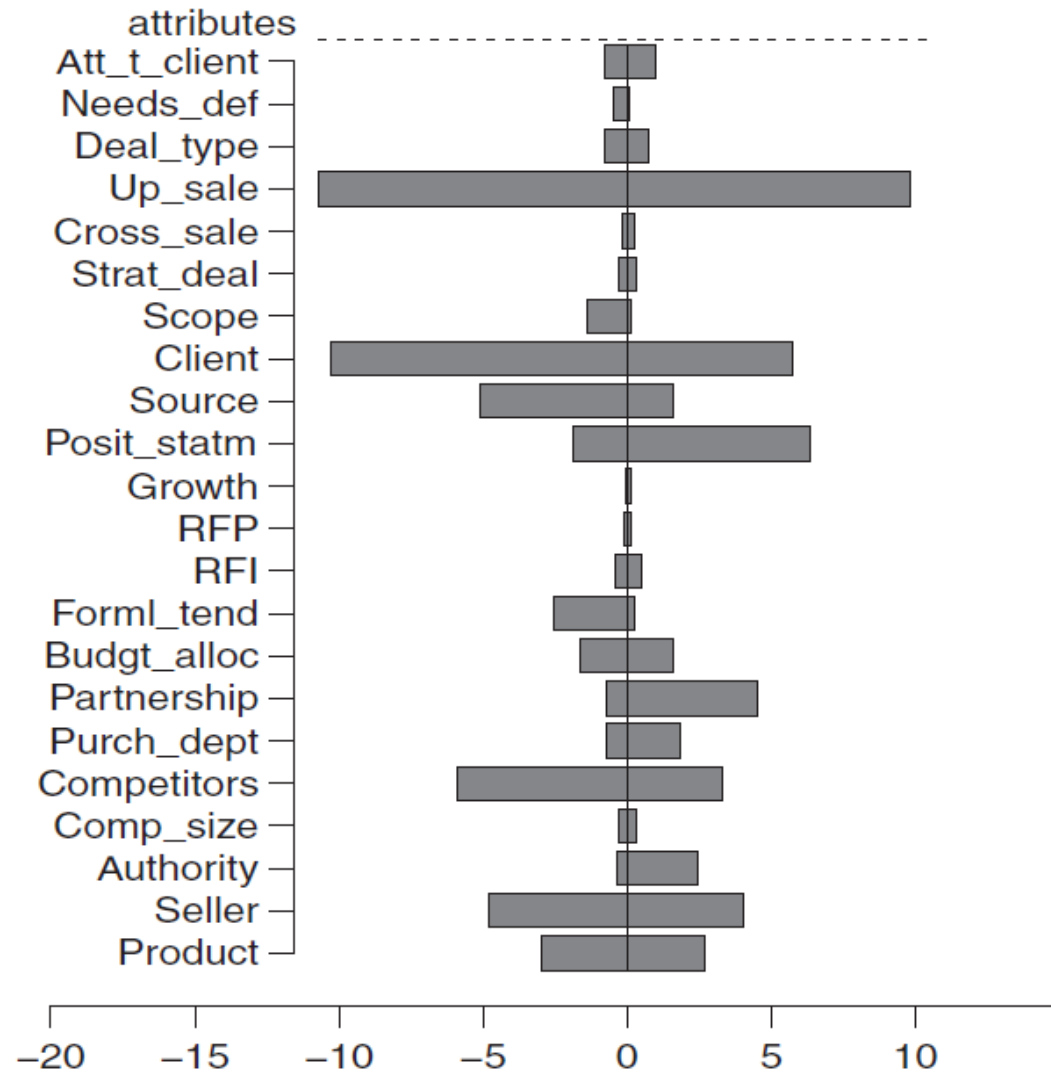




# B2B sales: EXPLAIN and IME



method EXPLAIN, type WE



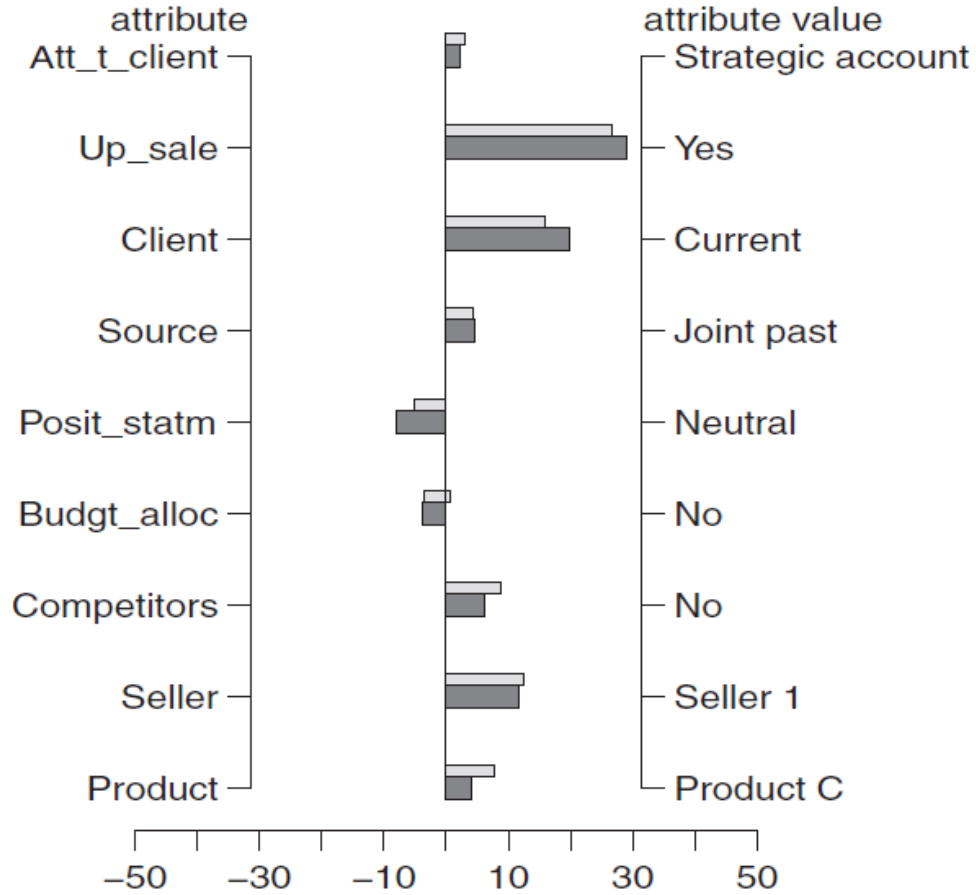
method IME





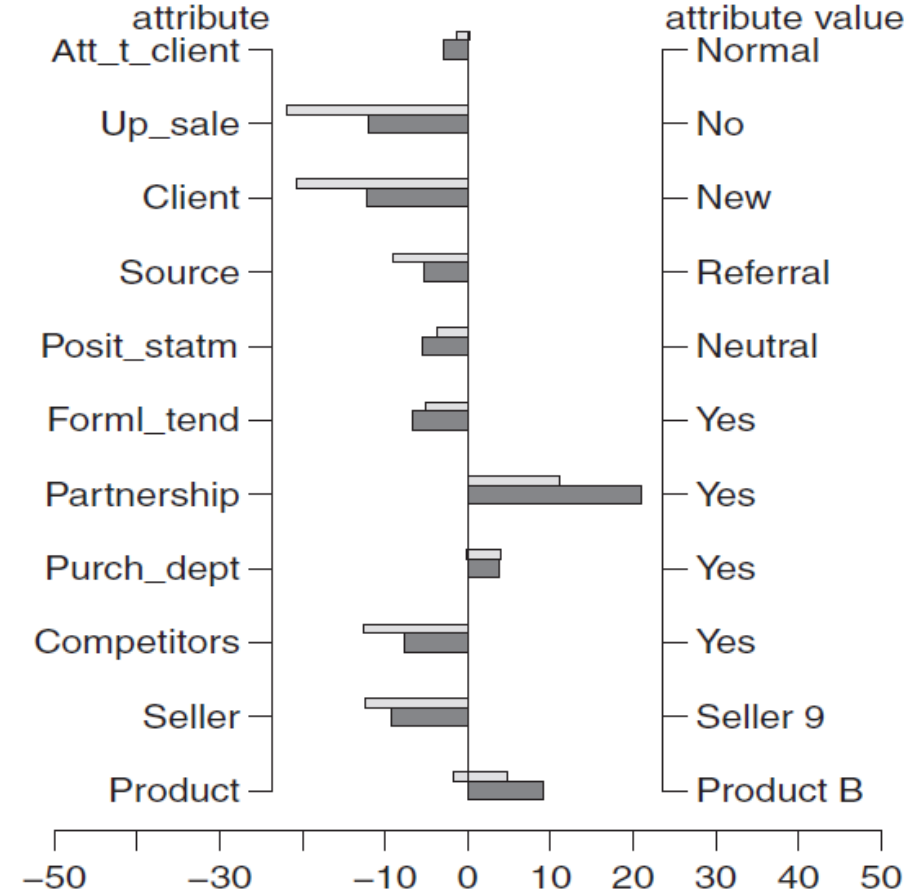
# B2B sales: learning from errors

**Explanation case, Status = Won**  
instance: 116, model: rf



method IME  
 $p(\text{Status}=\text{Won}) = 0.71$ ; true Status=Lost

**Explanation case, Status = Won**  
instance: 204, model: rf



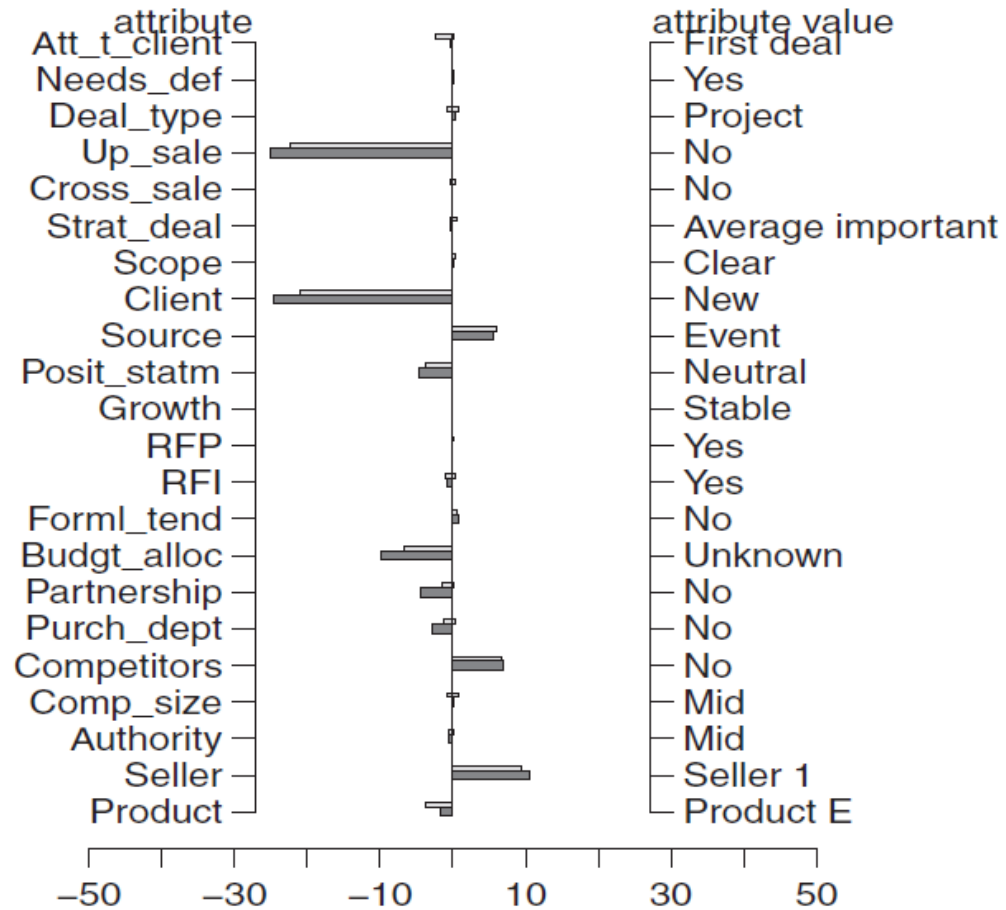
method IME  
 $p(\text{Status}=\text{Won}) = 0.38$ ; true Status=Won





# B2B: what if

What-if case, Status = Won  
instance: new, model: rf



method IME

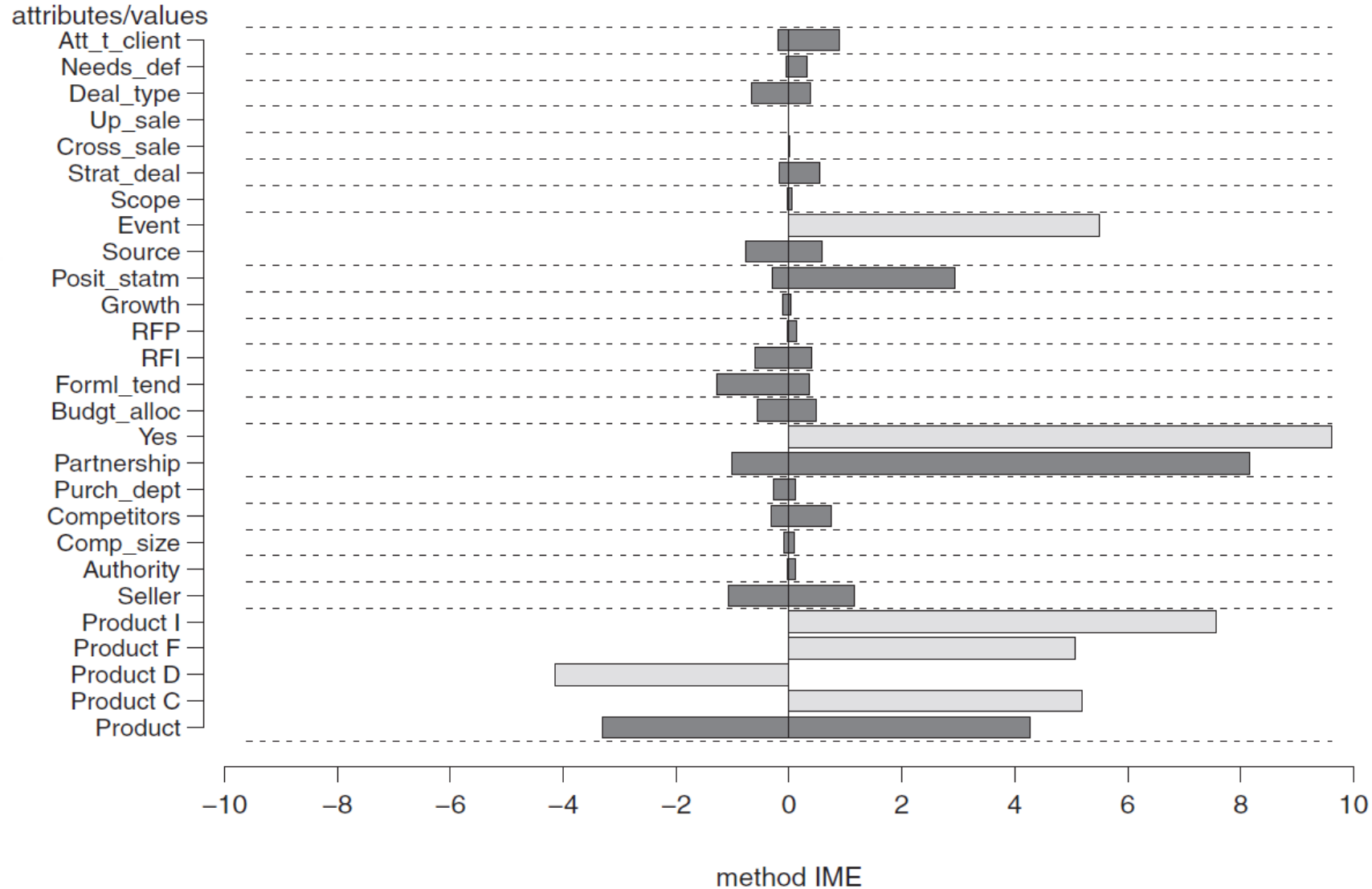
p(Status=Won) = 0.29; true Status=Open

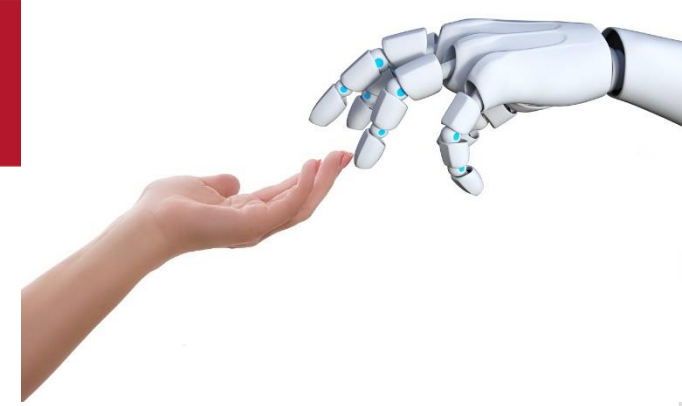




# B2B: change of distribution

Acquisition of new clients, Status = Won  
model: rf





# Lessons learned

- an effort needed to overcome the resistance
- human-in-the-loop is necessary to train, discuss, clean data, introduce explanations
- with increased use users gain trust in the methodology
- human mental models tend to be biased
- joint interactive approach beats both humans and ML models
- problem with slippages



# More related work

- symbolic models
  - straightforward comprehensible explanation for small models (decision trees, lists, rules)
- numeric models (mostly neural networks)
  - generation of symbolic models from generated additional instances
  - resulting models are large and incomprehensible
- explanations in form of nomograms for specific algorithm/model:
  - logistic regression, Naive Bayesian classifier, restricted decomposable kernels for SVM (
- visualization of SVM with a separating hyperplane in a restricted subspace
- explanation of neural networks based on propagation of gradients
- sensitivity analysis

# Opportunities

- better and more focused sampling
- better local explanation models
- interactions: detect and describe
- sequences: the order of attributes is important!
- images: decision areas, super-pixels
- better visualizations: human cognitive limitations



# Conclusions

- many successful approaches but
- lots of opportunities for improvements
- legal and practical need for explanations of ML models



# Learning with imbalanced data?

- supervised learning, classification setting
- at least one class is under-represented relative to others
- easy to get high classification accuracy, but this is not what we want

# Some motivational examples

- fraud detection (credit card, insurance, stock market)
- (many) medical diagnostics
- rare diseases
- bioinformatics (translation initiation site in DNA sequence...)
- response rate in direct marketing
- oil spills in satellite images
- industrial processes fault monitoring
- document filtering

# An example :

Negative class:

$N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$

$\boldsymbol{\mu}_1 = [0, 0]$

$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

200 instances

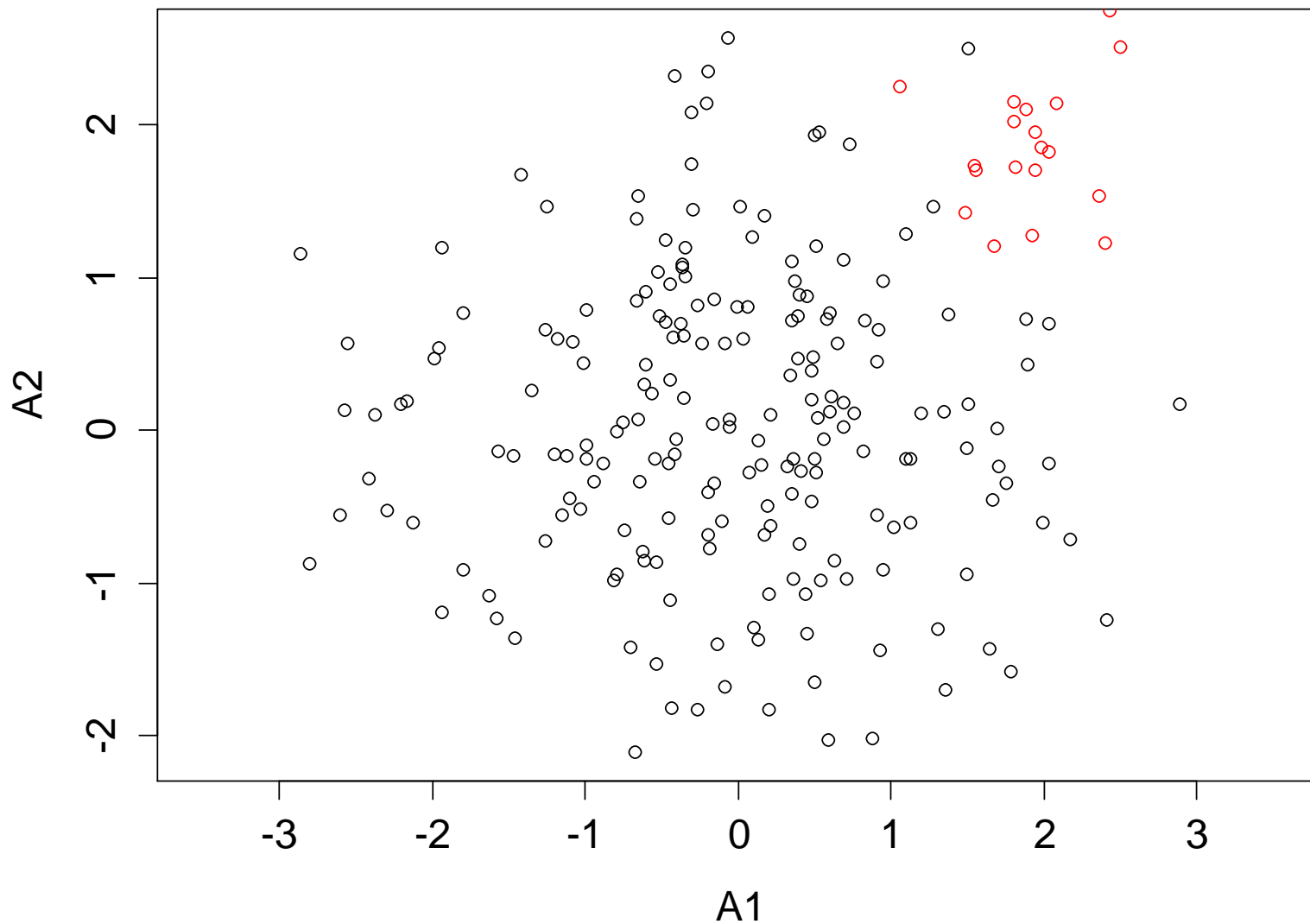
Positive class

$N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$

$\boldsymbol{\mu}_2 = [2, 2]$

$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$

20 instances



# An example

Negative class:

$N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$

$\boldsymbol{\mu}_1 = [0, 0]$

$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

200 instances

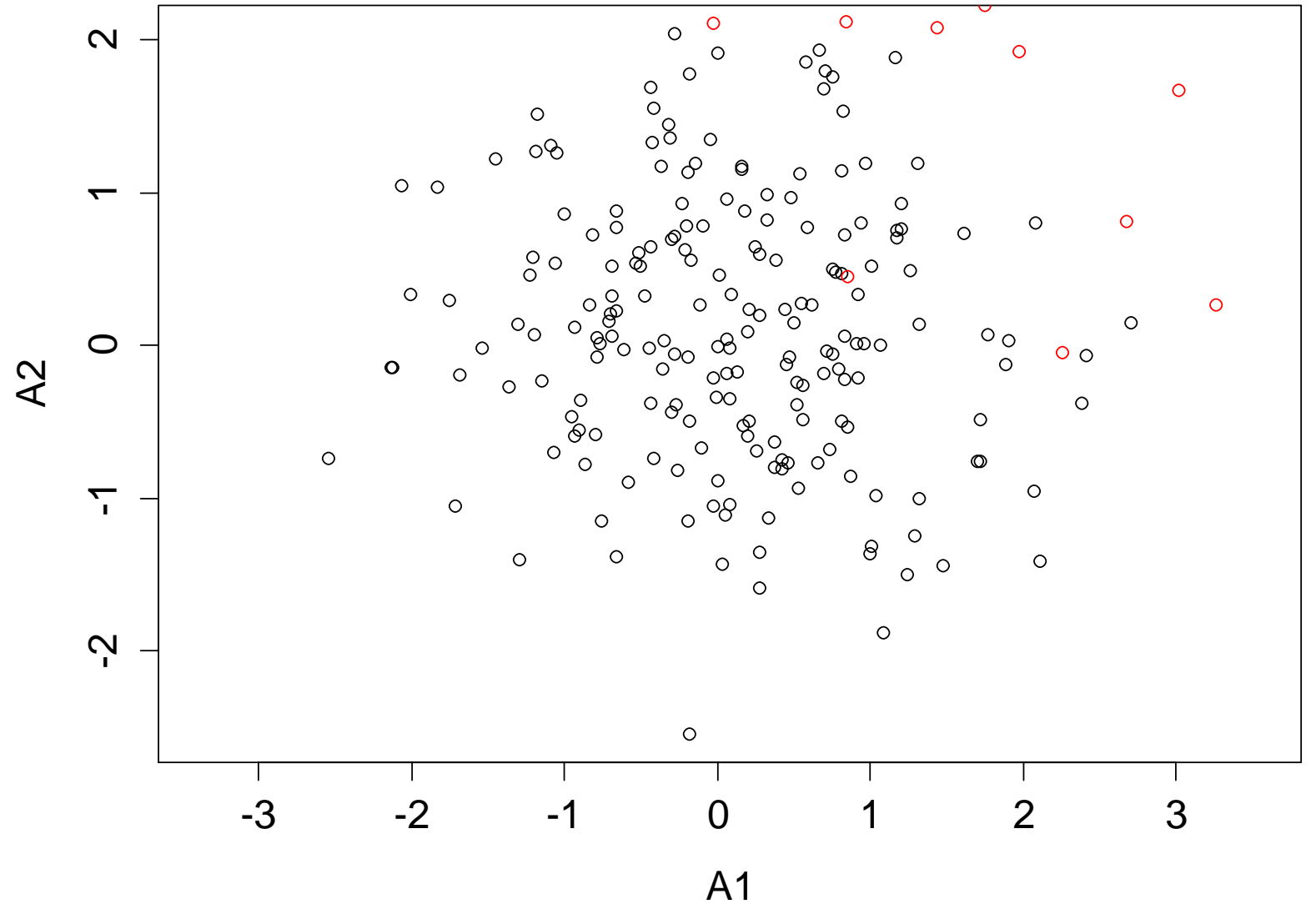
Positive class

$N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$

$\boldsymbol{\mu}_2 = [2, 2]$

$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

20 instances



# An example 3

Negative class:

$N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$

$\boldsymbol{\mu}_1 = [0, 0]$

$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

200 instances

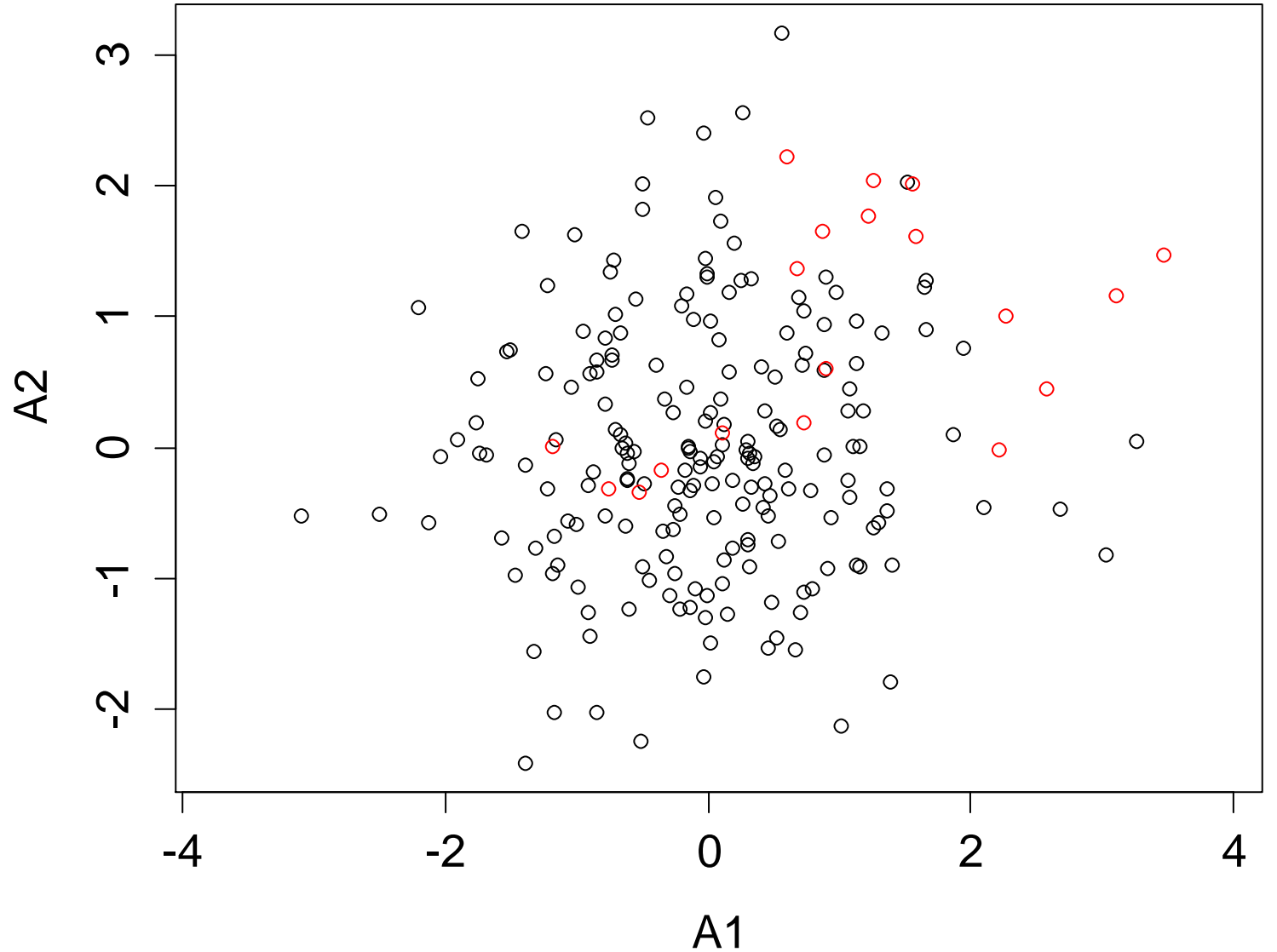
Positive class

$N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$

$\boldsymbol{\mu}_2 = [1, 1]$

$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

20 instances



# An example 4

Negative class:

$N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$

$\boldsymbol{\mu}_1 = [0, 0]$

$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

200 instances

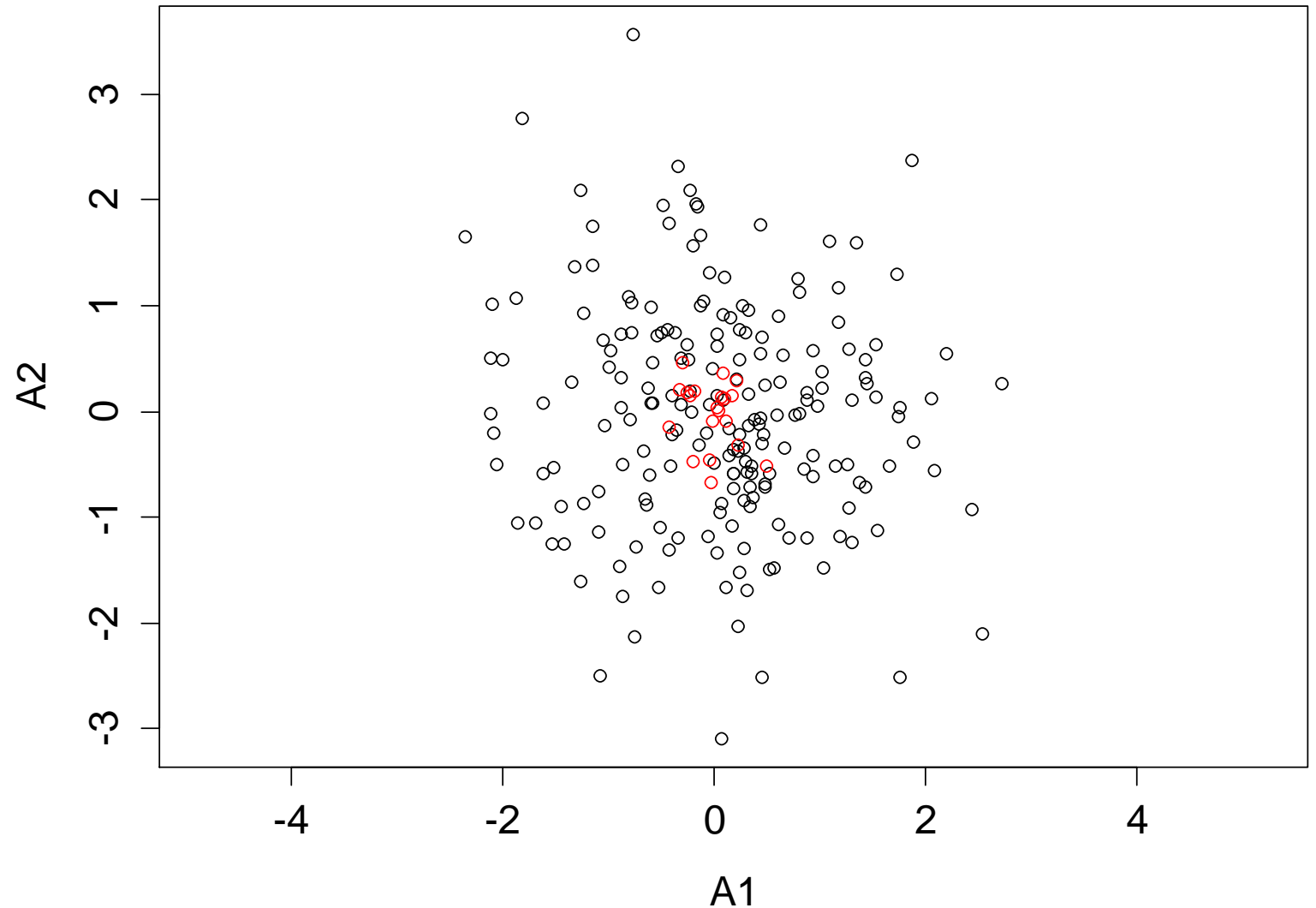
Positive class

$N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$

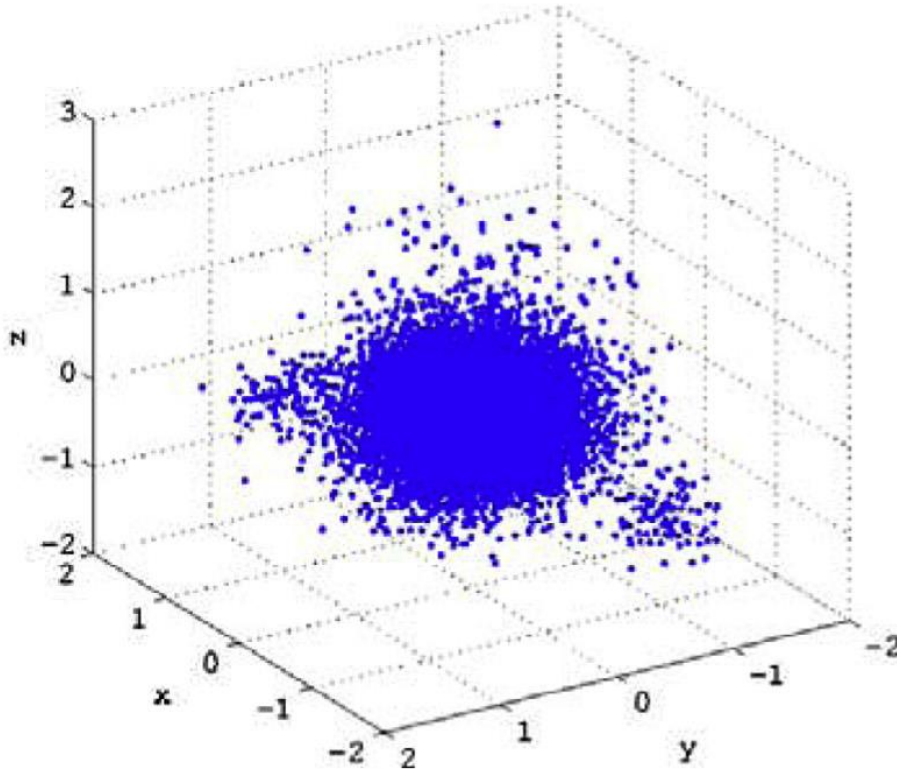
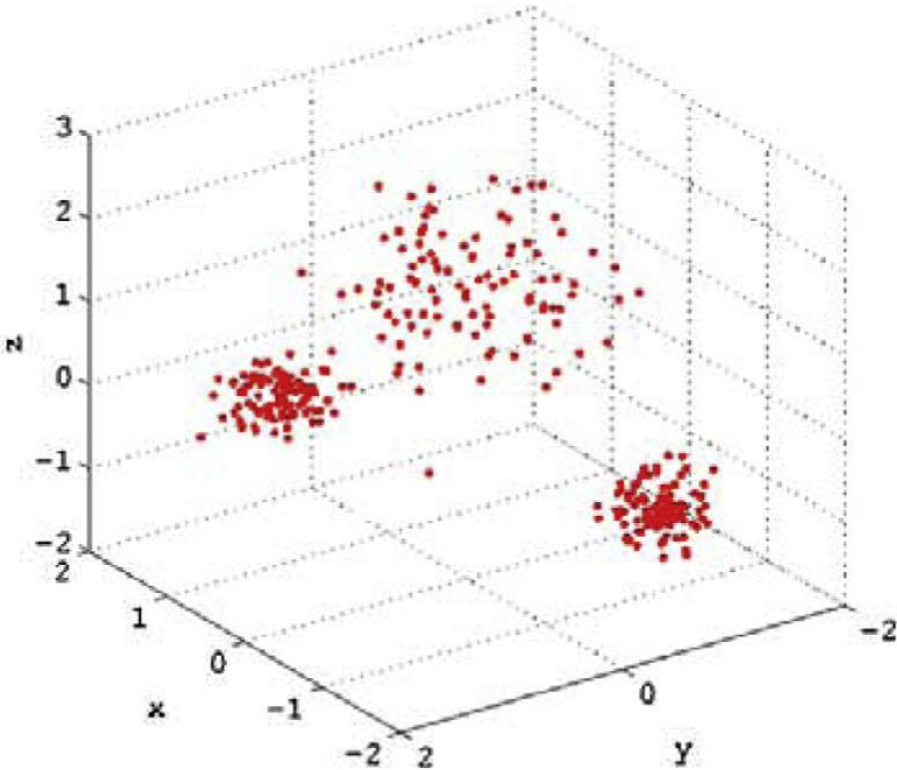
$\boldsymbol{\mu}_2 = [0, 0]$

$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$

20 instances



# An example 5





# Why is the problem difficult?

- standard learners are often biased towards the majority class
- classifiers reduce global quantities (e.g., error rate), not taking the data distribution into consideration.
- Result: examples from the majority class are well-classified, the minority class tend to be misclassified.
  
- small number of instances (absolute/relative rarity)
- many small subconcepts
- inappropriate classifiers (bias/variance)
- problem complexity
- inappropriate performance metrics

# Performance metrics

- misclassification matrix

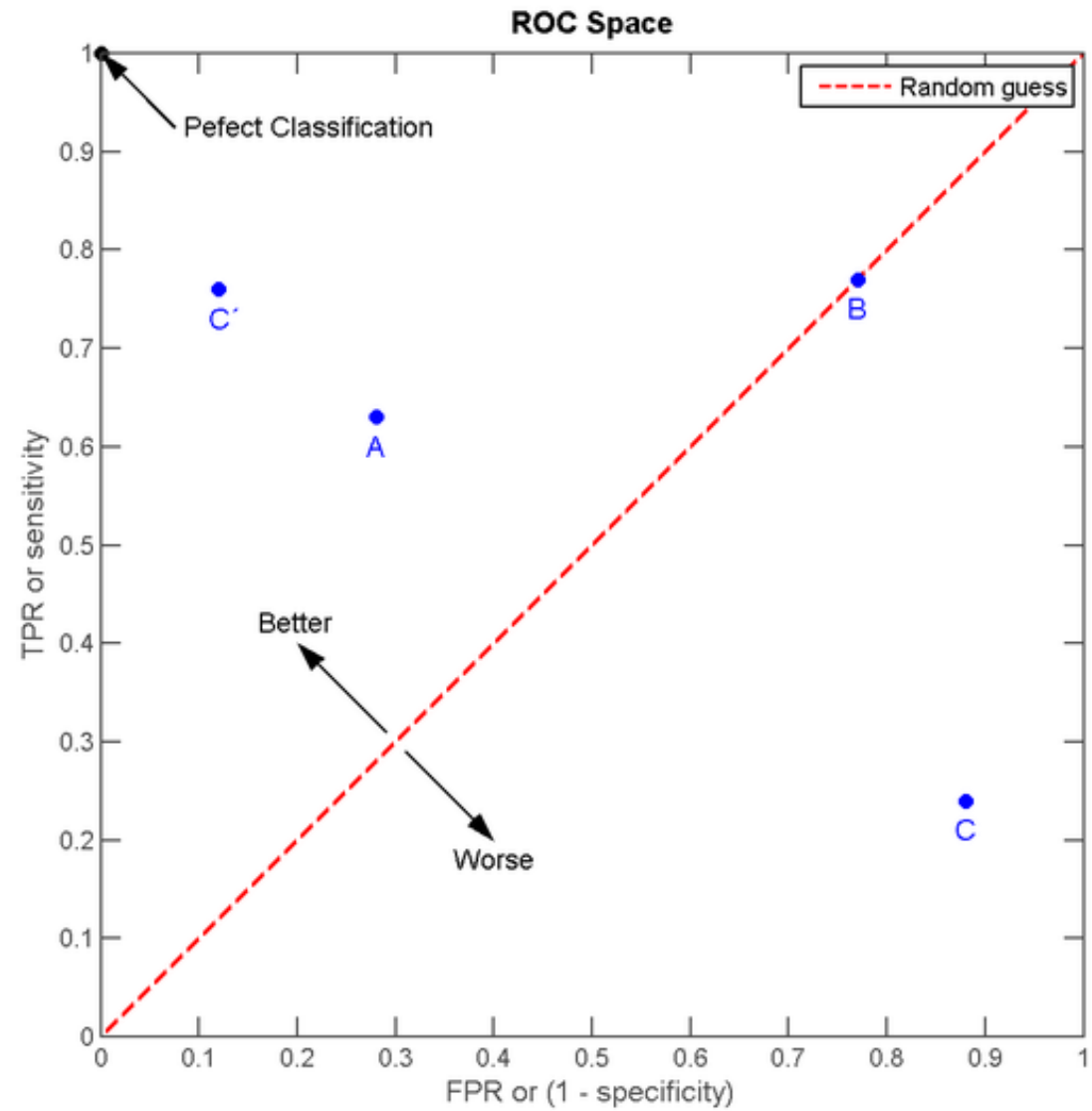
		Condition (as determined by "Gold standard")		
		Condition positive	Condition negative	
Test outcome	Test outcome positive	True positive	False positive (Type I error)	Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$
	Test outcome negative	False negative (Type II error)	True negative	Negative predictive value = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
		Sensitivity = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	Specificity = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Accuracy

# Accuracy

- sensitive to class distribution
- (takes both columns into account)
- class relative analysis does not make sense
- sensitivity, specificity are more appropriate

# ROC space

- AUC
- multiclass extensions



# Classical approaches

- Typical methods for imbalanced data in 2-class classification:
  - Oversampling: re-sampling of data from positive class
  - Under-sampling: randomly eliminate tuples from negative class
  - Threshold-moving: moves the decision threshold  $t$ , so that the rare class tuples are easier to classify, and hence, less chance of costly false negative errors

# Random sampling

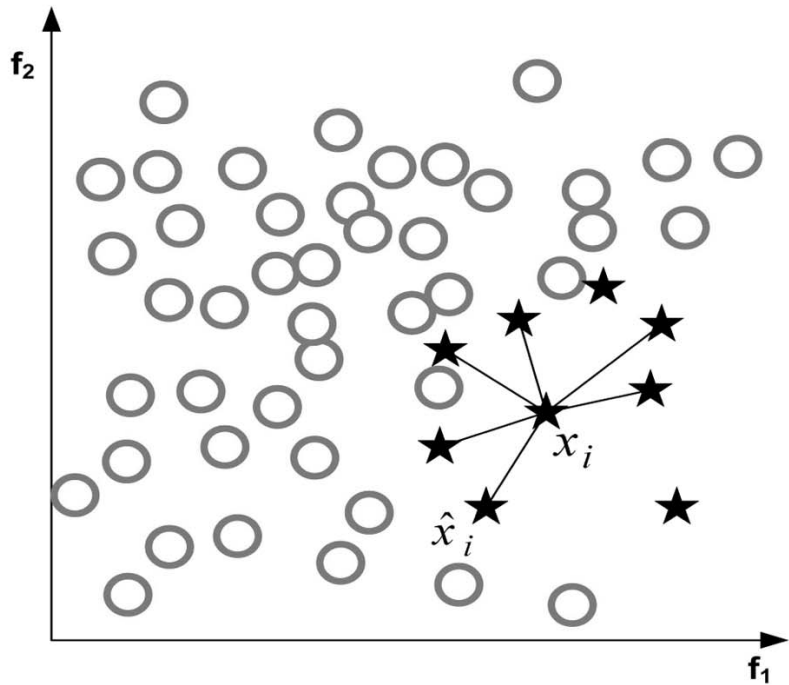
- random undersampling
  - randomly select a set of majority instances and remove them
  - problem: may miss some important subconcepts in the majority class
  - faster learning
- random oversampling
  - randomly select a set of minority instances, replicate them and add them to the learning set
  - problem: overfitting, slower learning

# Informed undersampling

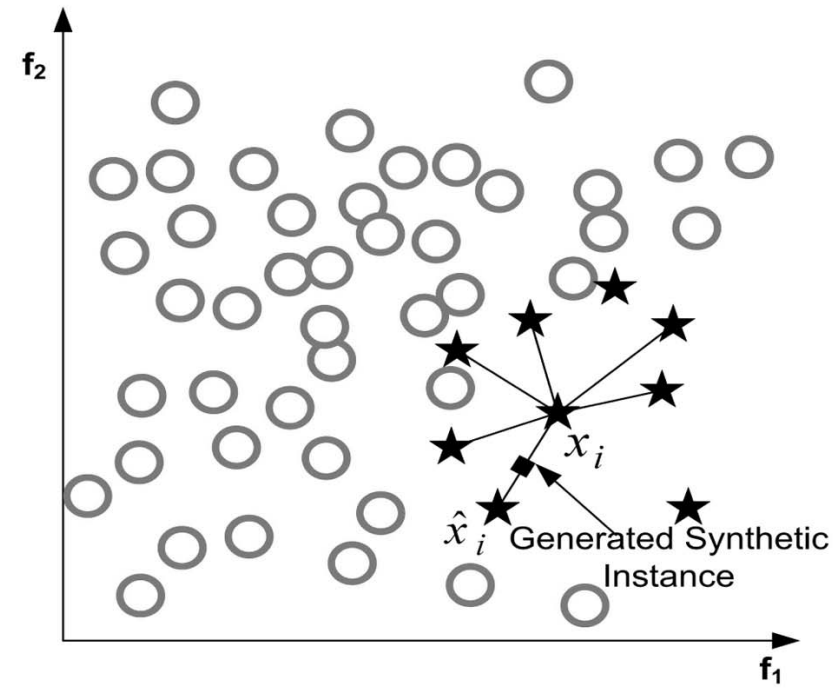
- K-nearest neighbor (KNN) based sampling
- several variants
  - select those majority instances whose average distance to three closest minority class examples is the smallest
  - select those majority instances whose average distance to three farthest minority class examples is the smallest
  - for each minority class example select a given number of closest majority class examples
  - ...
- KNN fails in high dimensional spaces

# Informed oversampling

- idea: create new similar minority class instances
- SMOTE (Synthetic Minority Oversampling Technique)
- $X_{\text{new}} = (X_i' - X_i) \cdot \delta$



(a)

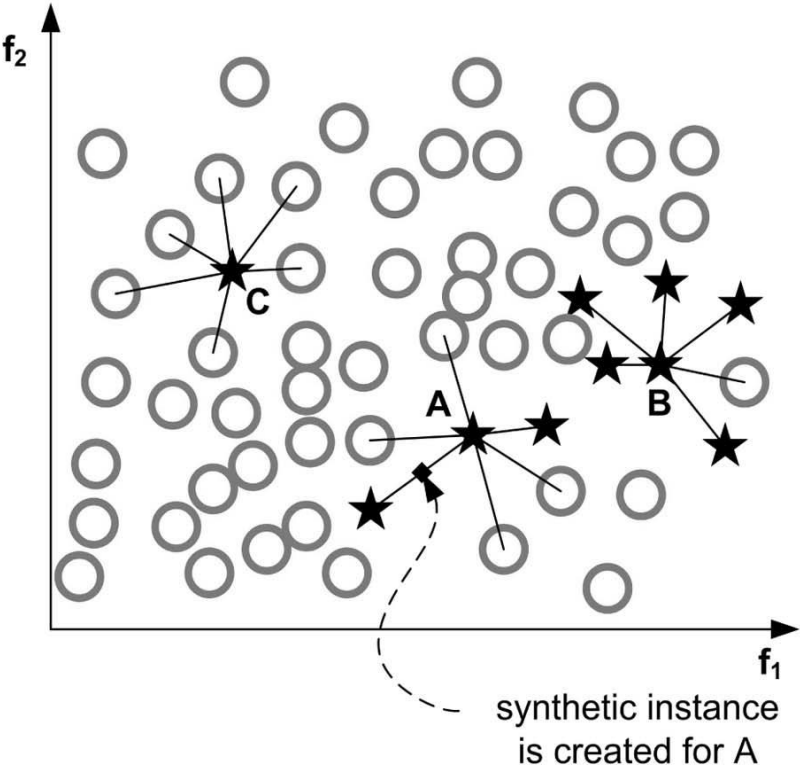


(b)



# SMOTE improvements

- problem of SMOTE: generates the same number of instances for each minority class instance disregarding its neighborhood
- Borderline-SMOTE: count overlap for KNN, generate only DANGER instances

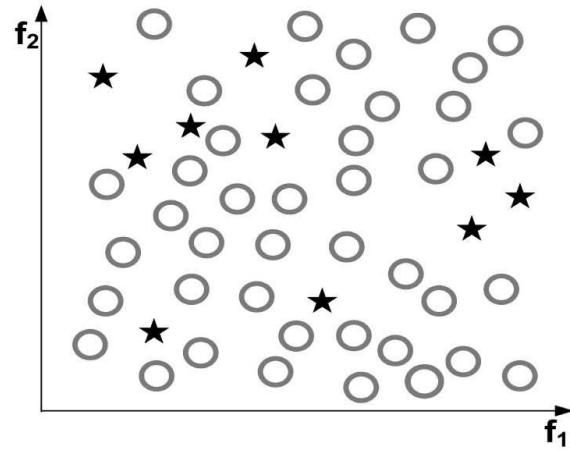


Consider 6-nearest neighbor:  $m=6$

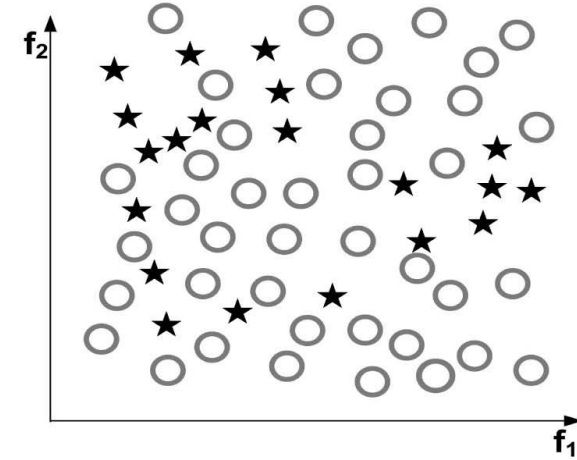
- For A: number of minority instance: 2  
number of majority instance: 4 → “DANGER” instance
- For B: number of minority instance: 5  
number of majority instance: 1 → “SAFE” instance
- For C: number of minority instance: 6  
number of majority instance: 0 → “NOISE” instance

# Oversampling with data cleaning

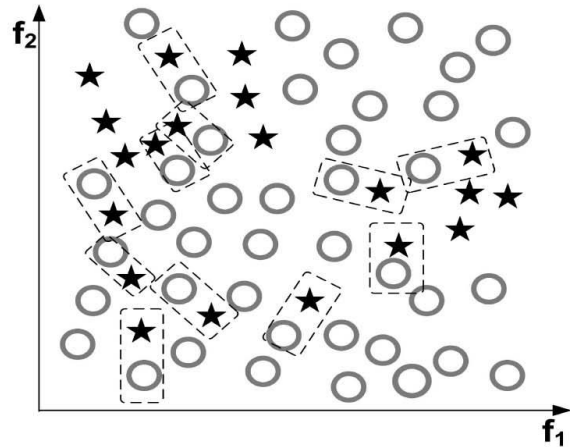
- SMOTE with removed Tomek links



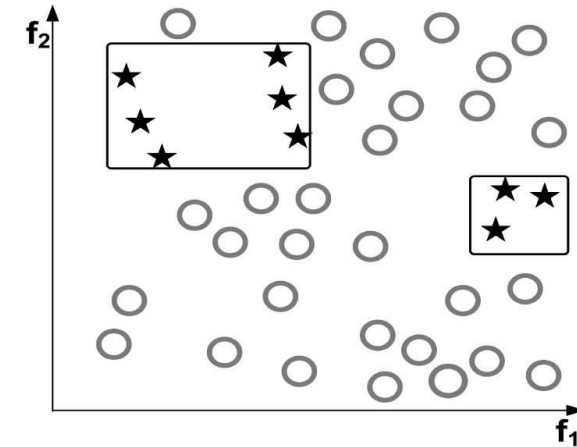
(a)



(b)

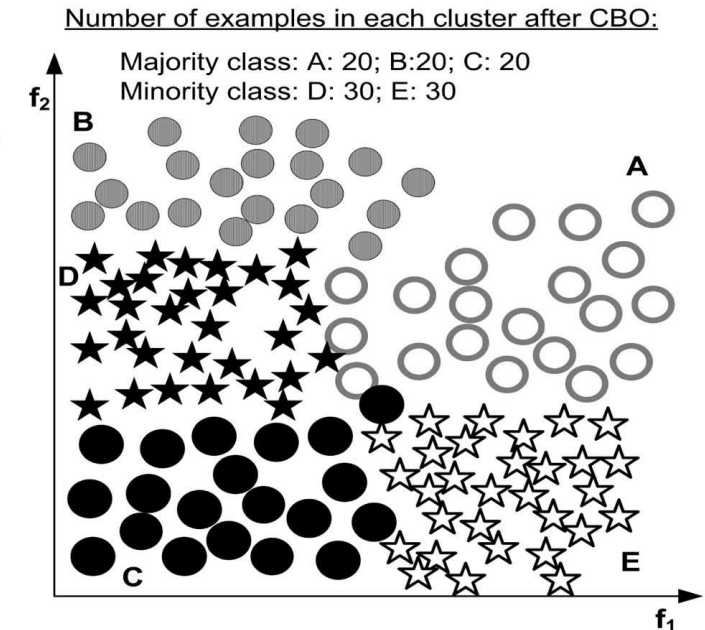
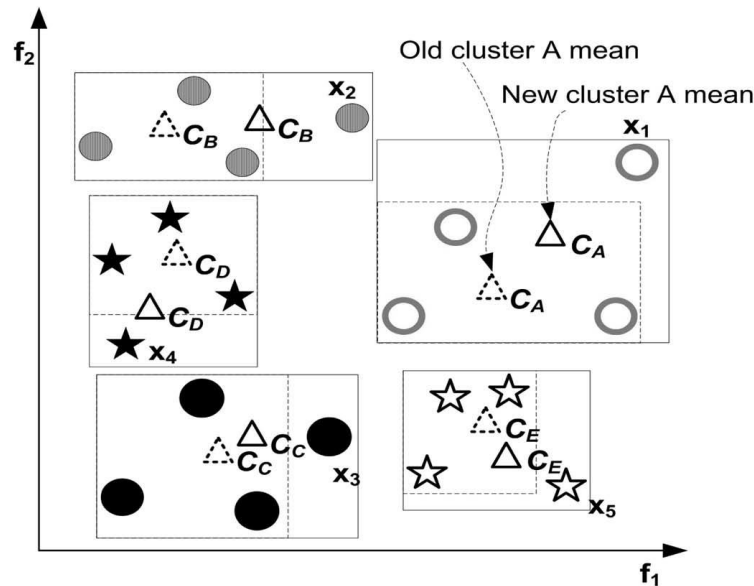
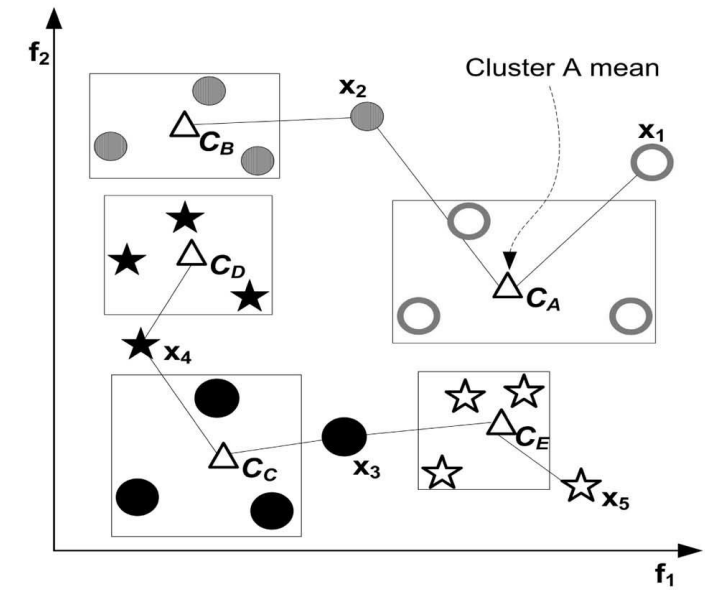
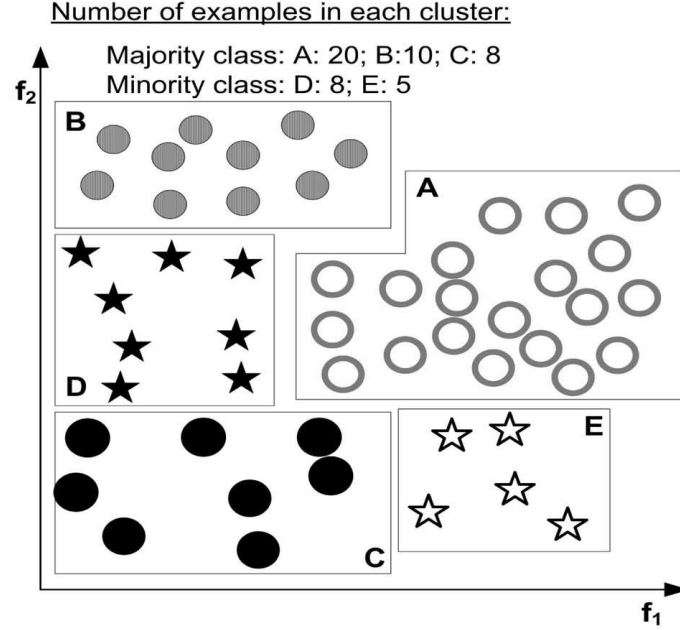


(c)



(d)

# Clustering based sampling



# Sampling combined with boosting

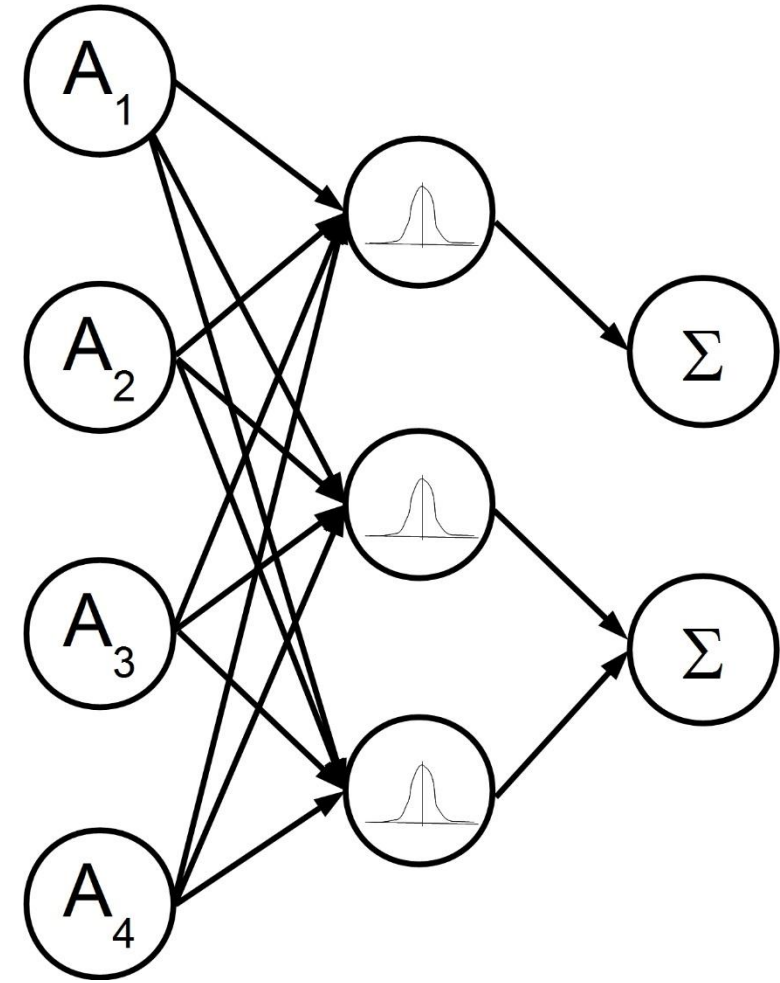
- Boosting idea: increase weight of the misclassified instances and iterate learning
- SMOTEboost: in each round the weights of minority class instances are increased using SMOTE
- EasyEnsemble:
  - create multiple majority class samples  $N_i$  of the same size as minority class examples  $P$
  - train several boosting models each with  $(N_i, P)$  as the training set
  - final model is a combination of all partial AdaBoost ensemble models
  - a bagged combination of boosted models

# BalancedCascade

- guided deletion
  - select a majority class sample  $N_i$  of the same size as minority class examples  $P$
  - train boosting model with  $(N_i, P)$  as the training set
  - delete correctly classified majority class examples from  $N$  and repeat
  - final model is a combination of all partial AdaBoost ensemble models

# Generating semi-artificial data

- idea:  
Use Radial Basis Function (RBF) network to learn properties of the data
- RBF learns a set of Gaussian kernels
- Gaussian kernels can be used in a generative mode to generate new data using variance matrix decomposition
- discrete data



# Using the generator

- generator performs implicit clustering
- data from each kernel can be generated independently and proportionally to the desired class distribution
- performance on original and generated data on average comparable
- use: in development, for small data sets, in simulations, preserve privacy, smooth data
- problems: very high dimensional data
- R package semiArtificial

# Cost-sensitive learning

- misclassification cost are NOT equal
- cost-sensitive problems are usually the ones with imbalanced class distribution
- costs (benefits) usually presented with cost matrix  $C$
- $C(i,j)$  is a cost of classifying class  $i$  as class  $j$
- optimal prediction selects class which minimizes expected loss



# Cost matrixes

- not all cost matrixes make sense

$$C3: \begin{bmatrix} 0 & 1 & 1 \\ 5 & 0 & 1 \\ 20 & 5 & 0 \end{bmatrix} \quad C4: \begin{bmatrix} 0 & 1 \\ 20 & 0 \end{bmatrix}$$

$$C5: \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 5 & 4 & 0 & 1 & 1 \\ 10 & 9 & 6 & 0 & 1 \\ 100 & 99 & 96 & 91 & 0 \end{bmatrix}$$

detecting exception, progressive health risk, financial loss

# Optimal classification with costs

- minimizing risk
- risk  $R(c_i|x) = \sum_{j=1}^c P(c_j|x)C(c_i, c_j)$
- crucial: good estimation of probabilities  $P(c_j|x)$
- calibration of probabilities

# MetaCost

- learn an ensemble with bagging
- relabel each instance according to the

$$\mathit{arg\ min}_i \sum_{j=1}^c P(c_j|x) C(c_i, c_j)$$

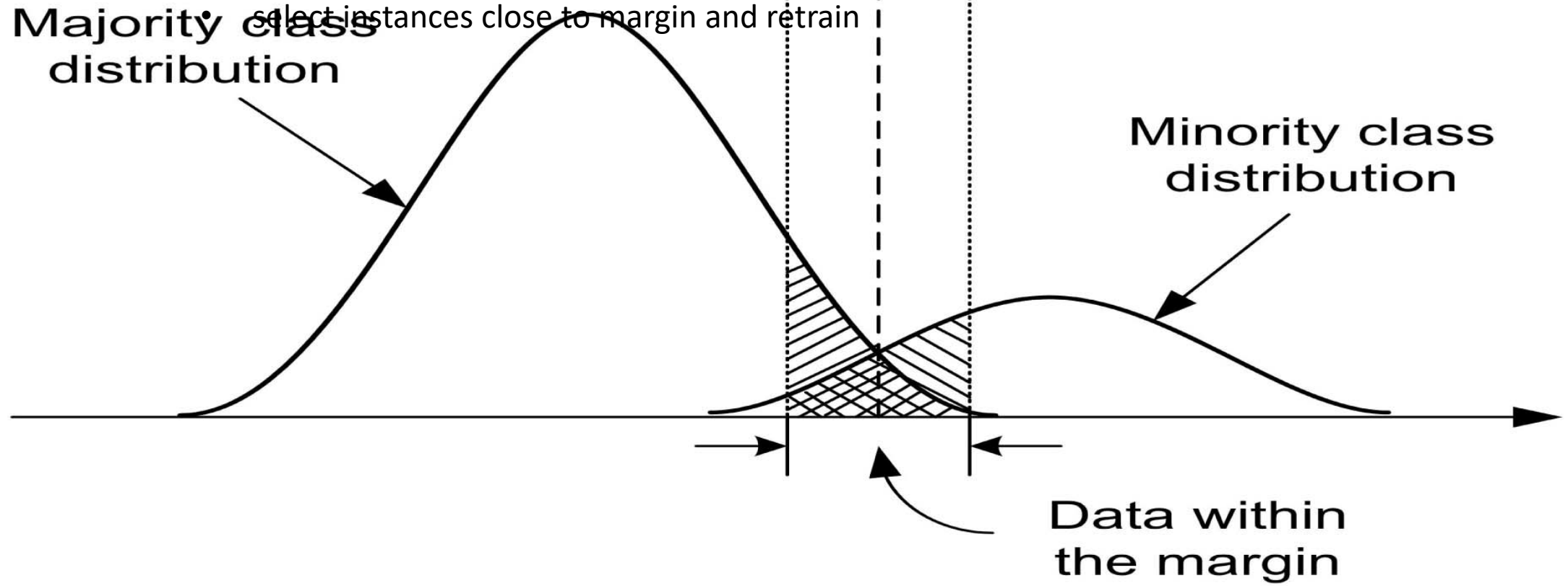
- $p(c_i|x)$  are obtained with bagging
- relearn with new class labels

# Integrating cost into learners

- Boosting integrates cost information through instance reweighting (AdaCost)
- SVM integrates cost by cost weighted margin or SMOTE-based sampling
- neural networks integrate cost into error function (used in probabilistic estimates, backpropagation, learning rate, output)
- cost-sensitive decision trees:
  - decision threshold
  - attribute evaluation criterion
  - pruning of trees

# Active learning

- in SVM setting class imbalance close to margin is much lower  
select instances close to margin and retrain

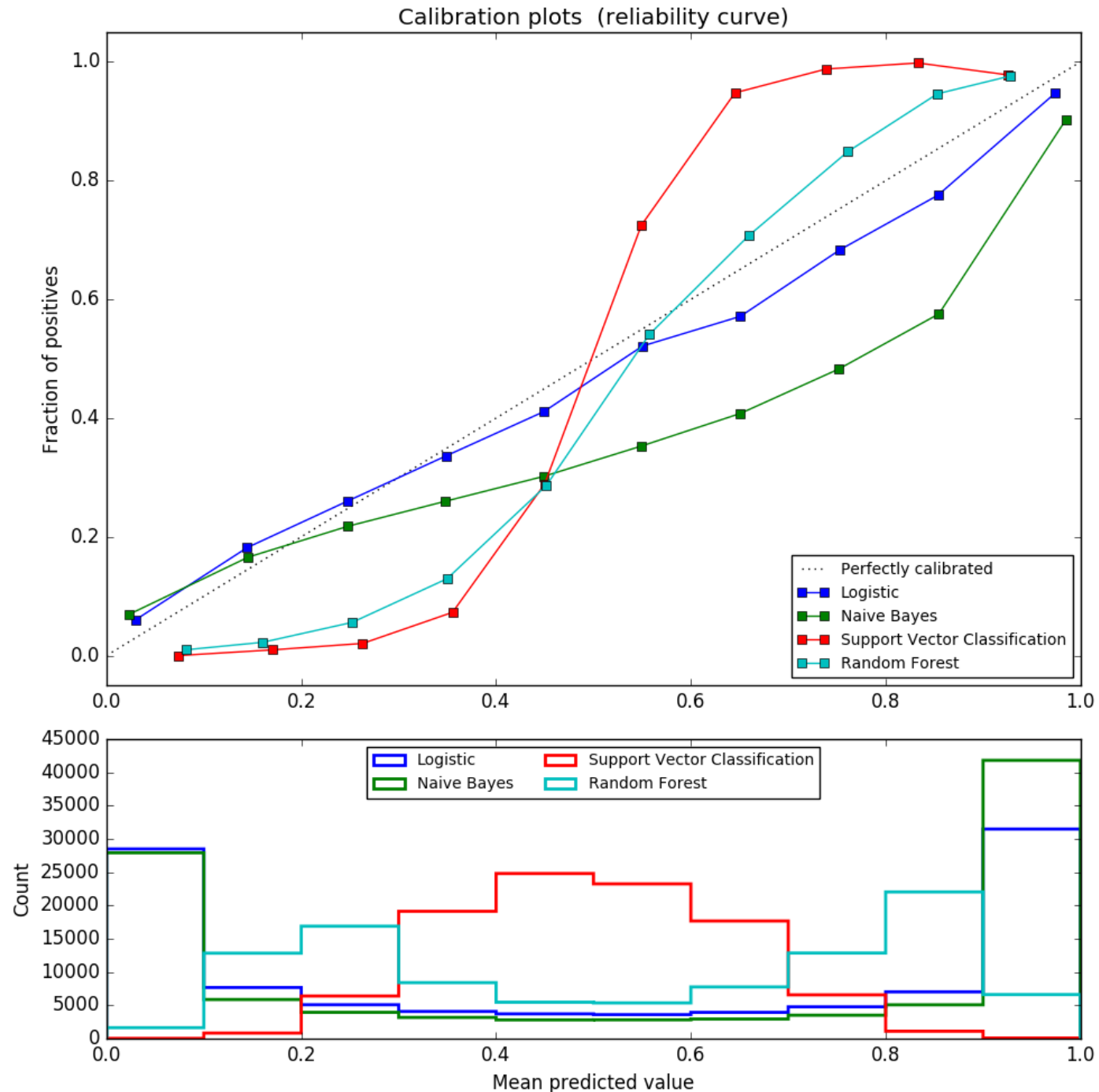


# Obtaining good probabilities

- When performing classification you often want not only to predict the class label, but also obtain a probability of the respective label.
- This probability gives you some kind of confidence on the prediction.
- Some models can give you poor estimates of the class probabilities and some even do not support probability prediction.
- To obtain reliable probabilities from model's scores one has to calibrate it.
- Well calibrated classifiers are probabilistic classifiers for which the output of the method can be directly interpreted as a confidence level. For instance, a well calibrated (binary) classifier should classify the samples such that among the samples to which it gave a predicted value close to 0.8, approximately 80% actually belong to the positive class.

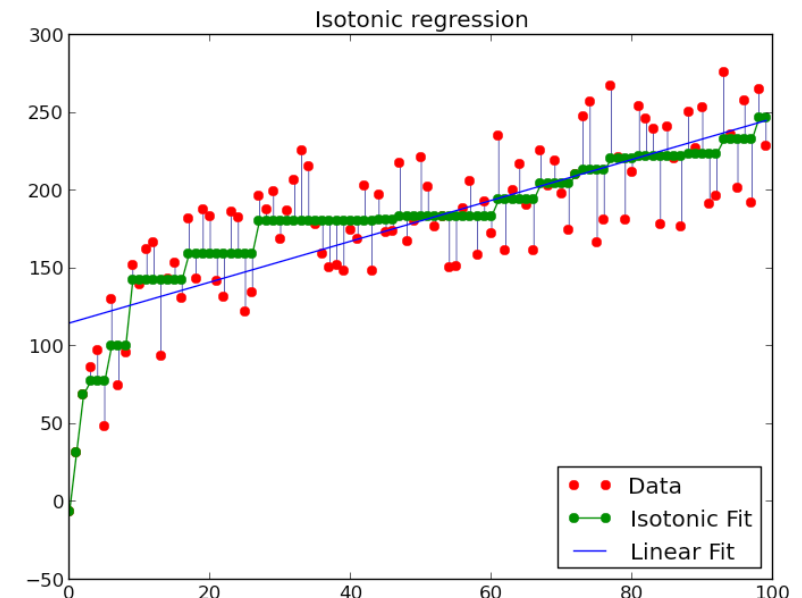
# Reliability graphs

- Reliability graphs show how predicted (horizontal axis) and actual (vertical axis) probabilities relate to one another.
- Ideally probabilities would be placed on the diagonal  $y = x$  line, meaning that for each band the proportion of event realizations would match the predicted probabilities.
- Left graph shows typical behavior of some classifiers.



# Calibration algorithms

- Note: a separate calibration data set is required
- The most popular calibration methods are
  - binning histogram method, where we give number and/or position of splits in advance and then fix each bin with the correct value
  - Platt's scaling: performing logistic regression on the output of the model with respect to the true class labels.
  - isotonic regression, which automatically generates splits based on the distribution of predicted probabilities and actual results. We fit a piecewise-constant non-decreasing function instead of logistic regression. Piecewise-constant non-decreasing means stair-step shaped.





# Ethical concerns: as individual and in society

- Ubiquitous Data Mining
  - Data mining is used everywhere, e.g., online shopping
  - Example: Customer relationship management (CRM)
- Invisible Data Mining
  - Invisible: Data mining functions are built in daily life operations
  - Ex. Google search: Users may be unaware that they are examining results returned by data mining
  - Invisible data mining is useful and desirable?
  - Invisible mining needs to consider privacy, efficiency and scalability, user interaction, incorporation of background knowledge and visualization techniques, finding interesting patterns, real-time, ...

# Privacy, security and social impacts of data mining

- Many data mining applications do not touch personal data
  - E.g., meteorology, astronomy, geography, geology, biology, and other scientific and engineering data
- Many DM studies are on developing scalable algorithms to find general or statistically significant patterns, not touching individuals
- The real privacy concerns:
  - unconstrained access to individual records, especially privacy-sensitive information
  - matching of records from different databases
- Solution 1: Removing sensitive IDs associated with the data
- Solution 2: Data security-enhancing methods
  - Multi-level security model: permit access to only authorized level
  - Encryption: e.g., *blind signatures*, *biometric encryption*, and *anonymous databases* (personal information is encrypted and stored at different locations)
- Solution 3: Privacy-preserving data mining methods

# Privacy-preserving data mining

- Privacy-preserving (privacy-enhanced or privacy-sensitive) mining:
  - Obtaining valid mining results without disclosing the underlying sensitive data values
  - Often needs trade-off between information loss and privacy
- Privacy-preserving data mining methods:
  - Randomization (e.g., perturbation): add noise to the data in order to mask some attribute values of records
  - K-anonymity and l-diversity: alter individual records so that they cannot be uniquely identified
    - k-anonymity: Any given record maps onto at least k other records
    - l-diversity: enforcing intra-group diversity of sensitive values
  - Distributed privacy preservation: data partitioned and distributed either horizontally, vertically, or a combination of both
  - Downgrading the effectiveness of data mining: the output of data mining may violate privacy
    - Modify data or mining results, e.g., hiding some association rules or slightly distorting some classification models

# Final note

- Data mining is like life: interesting, full of surprises, funny and messy. Enjoy it!

