Statistisches Data Mining (StDM)
Woche 9

Oliver Dürr

Institut für Datenanalyse und Prozessdesign

Zürcher Hochschule für Angewandte Wissenschaften

oliver.duerr@zhaw.ch

Winterthur, 15 November 2016

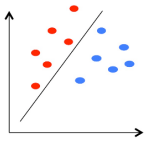
No laptops, no phones, no problems

Multitasking senkt Lerneffizienz:

- **Keine Laptops im Theorie-Unterricht Deckel zu oder fast zu (Sleep modus)**

Overview of classification (until the end to the semester)

Classifiers



K-Nearest-Neighbors (KNN)
Logistic Regression
Linear discriminant analysis
Classification Trees
Support Vector Machine (SVM)
Neural networks NN
Deep Neural Networks (e.g. CNN, RNN)
...

Evaluation

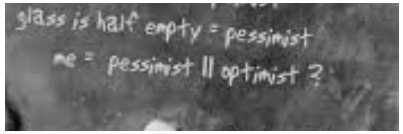
Cross validation
Performance measures
ROC Analysis / Lift Charts

Combining classifiers

Bagging
Boosting
Random Forest

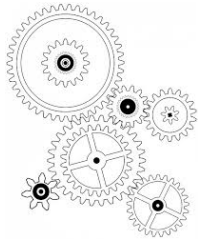
Theoretical Guidance / General Ideas

Bayes Classifier
Bias Variance Trade off (Overfitting)



Feature Engineering

Feature Extraction
Feature Selection



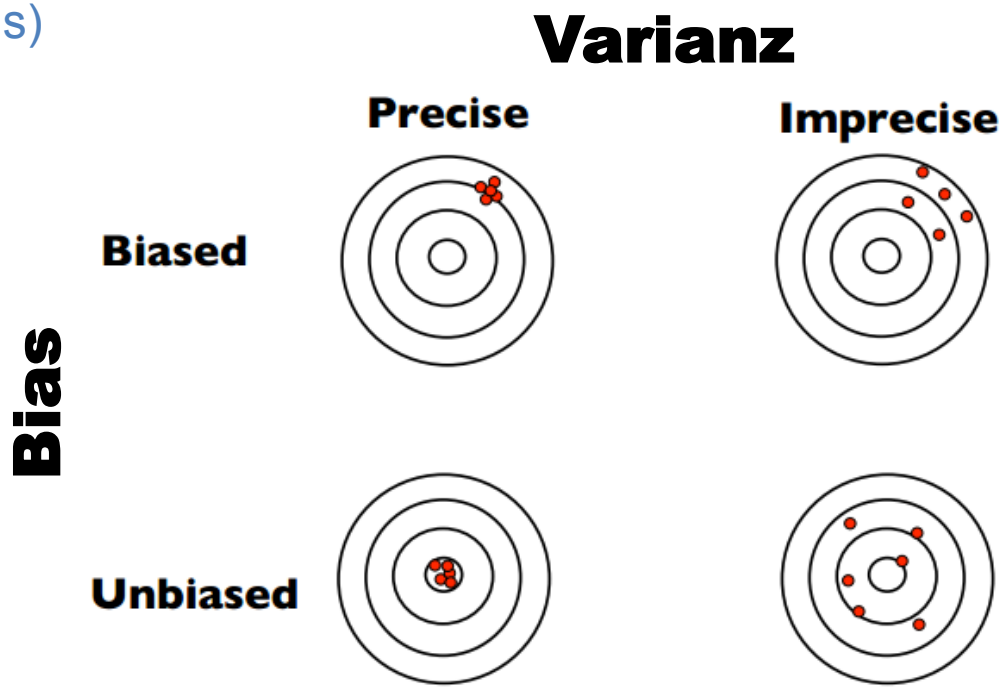
Crossvalidation

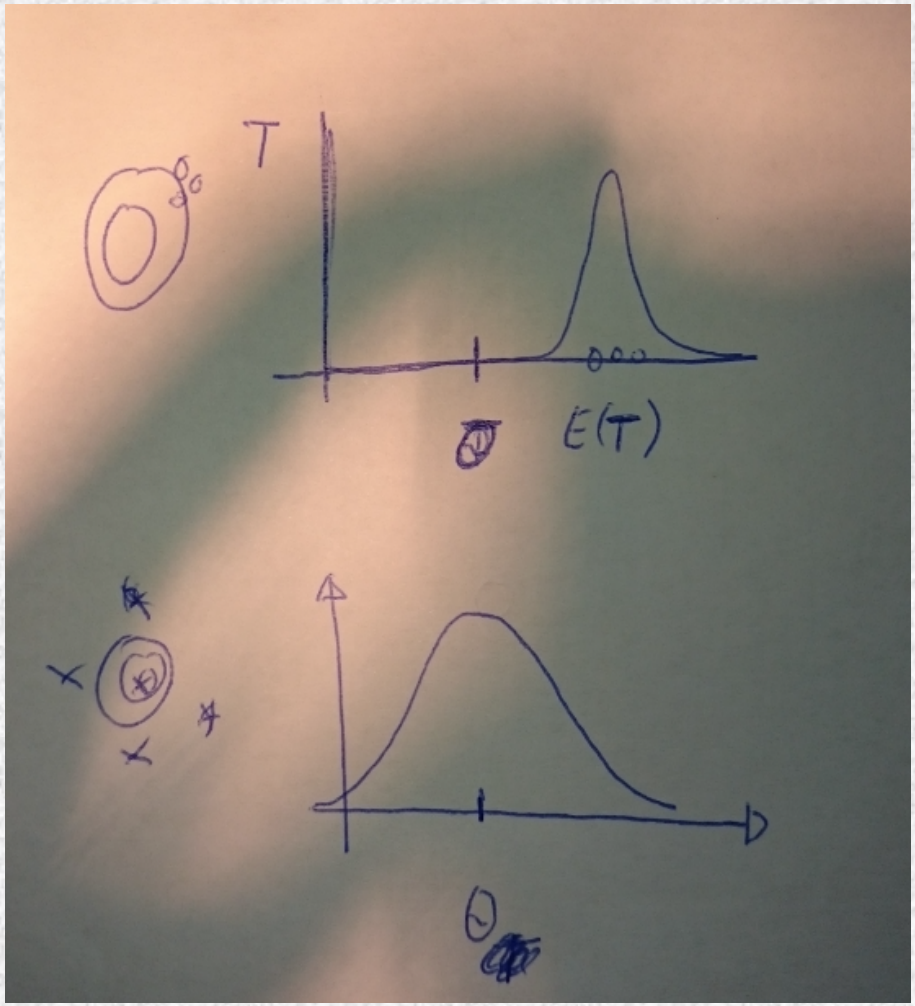
Outline

- Cross Validation (performance measures and splitting techniques)
  - Measures
    - Accuracy
  - Splitting in Training / Testset
    - The Validation Set Approach
    - Leave-One-Out Cross Validation
    - K-fold Cross Validation
    - Bias-Variance Trade-off for k-fold Cross Validation
  - More Measures
  - Pitfalls of cross validation approach

Reminder: Wast3 Estimation

- Goal:
  - Estimate the performance (e.g. accuracy) on unseen data “test-error” using only the training data
  - As any statistical estimation this estimator can have
    - Bias (systematic error)
    - Variance (fluctuations)





3 Grössen

VAR(T)
E(T)
MSE = E((T-O)^2)

Accuracy as performance measure

Evaluate prediction accuracy on data

Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

For an ideal classifier the off-diagonal entries should be zero: c=0, b=0, or Accuracy=1

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

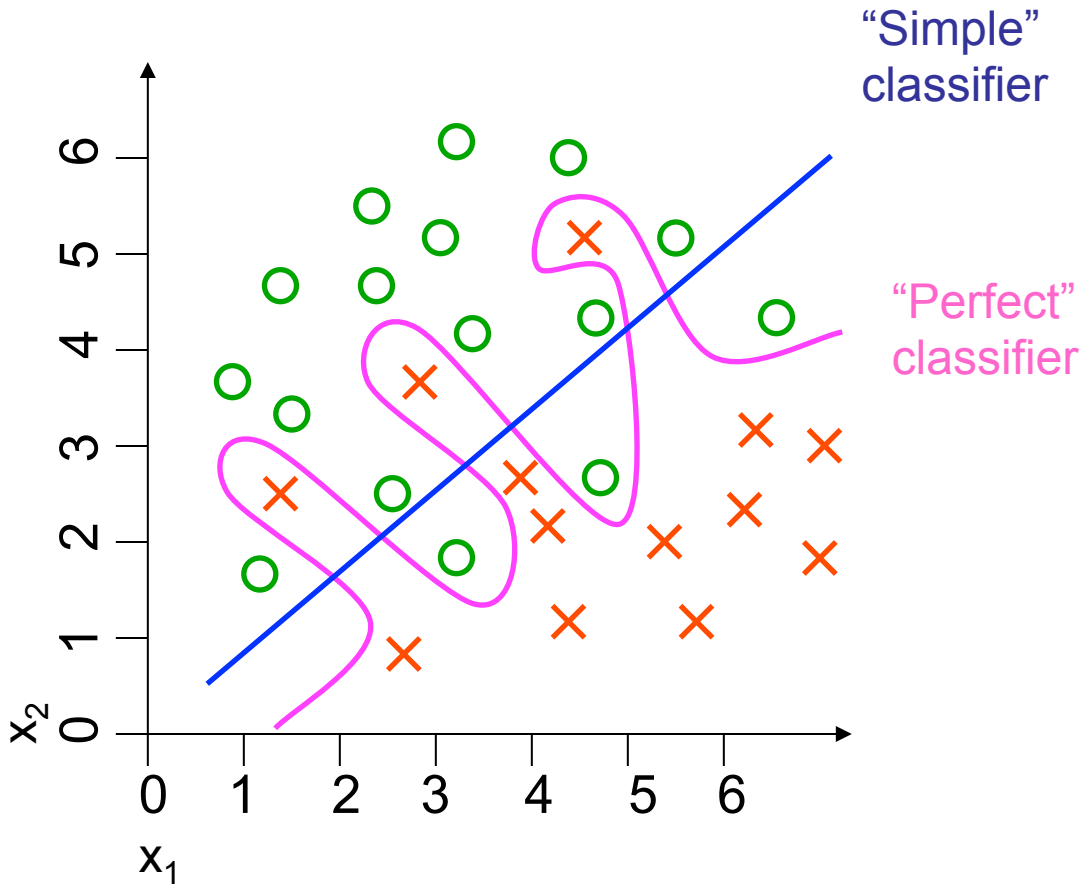
d: TN (true negative)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Simply count the # correct / all

Source: Tan, Steinbach, Kumar

"Perfect" Vs. "Simple" classifier



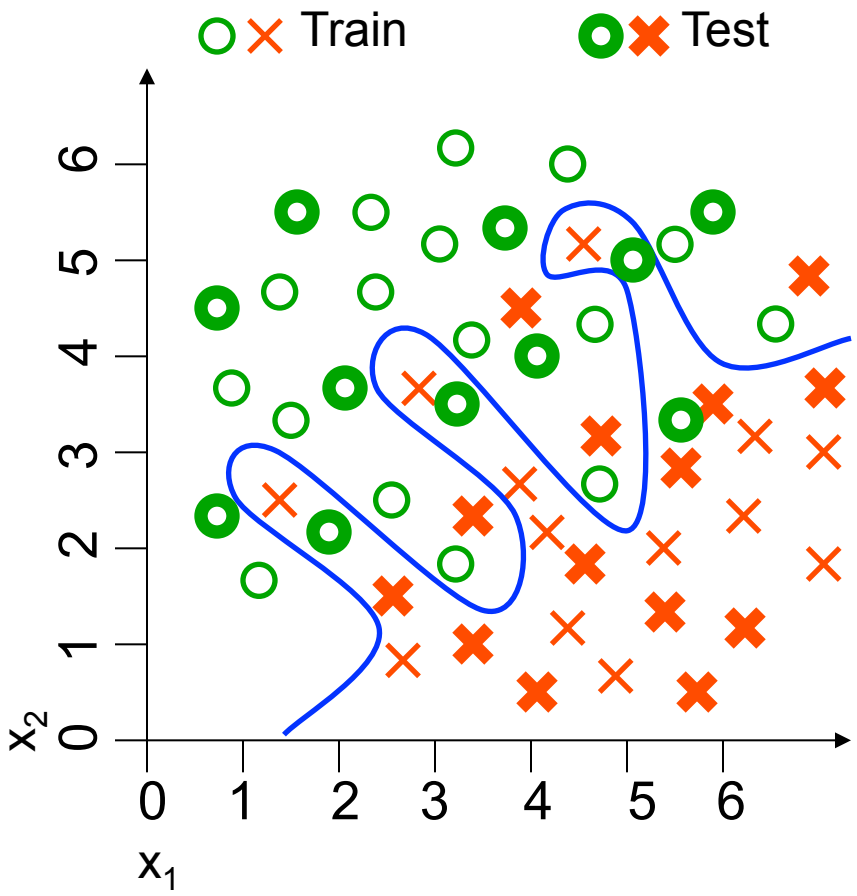
Which is better?

Check on a test-set (don't use all you labeled data to train)

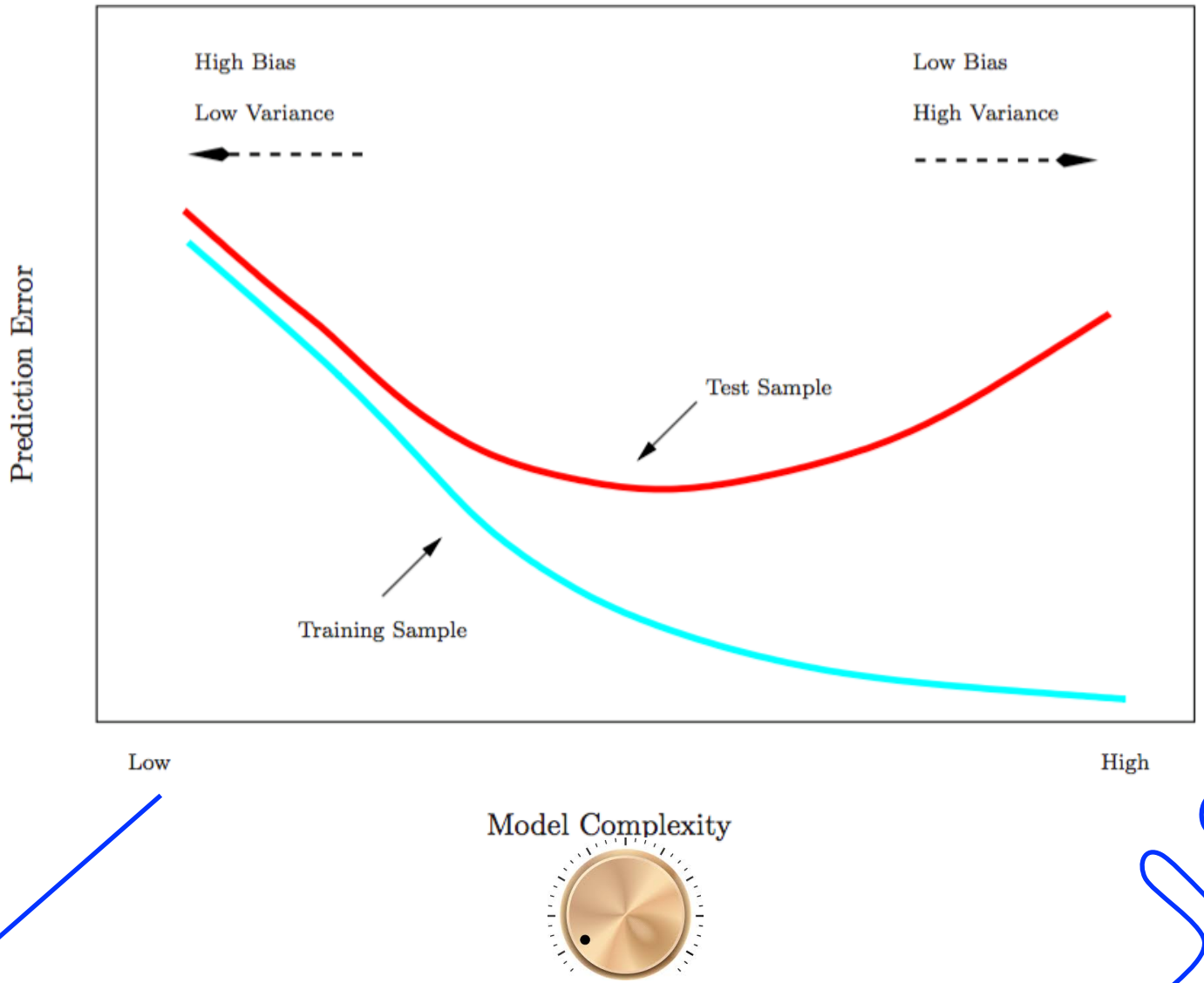
Cross validation of the "Perfect" classifier

Training set:
0%misclassification

Test set:
8/25=24%
misclassification



Typical Error Curve



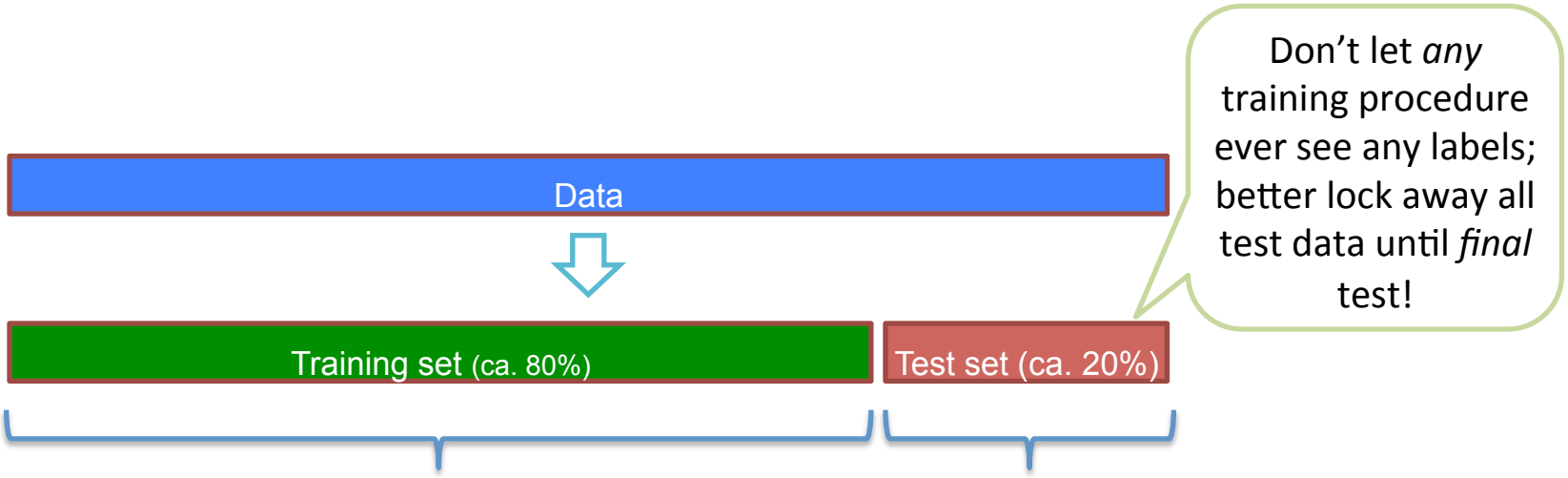
Questions for X-Validation

- Model Selection (which model to take)
  - Often there is a knob controlling the complexity of a classifier
    - Number of NN in KNN?
  - Number of features to use?
  - Square / log the features and add them?
  - Shall I use Logistic Regression or LDA?

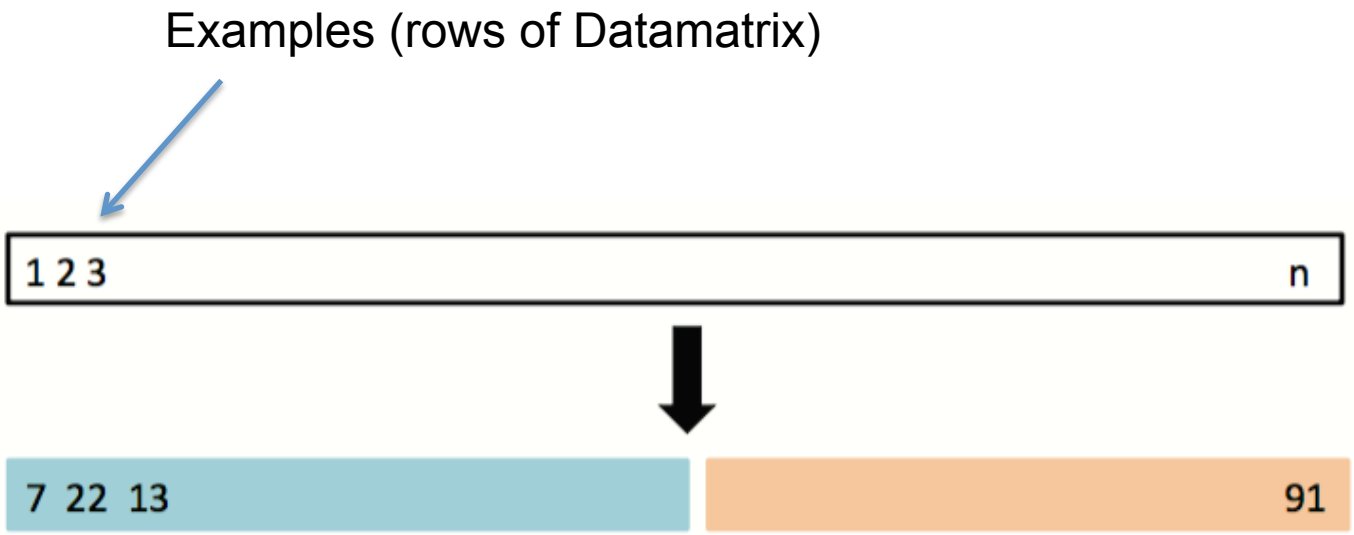
- Model Evaluation
  - How good is the performance on new unseen data.

How to be on the safe side

Typical strategy, spare some data for the final testing.



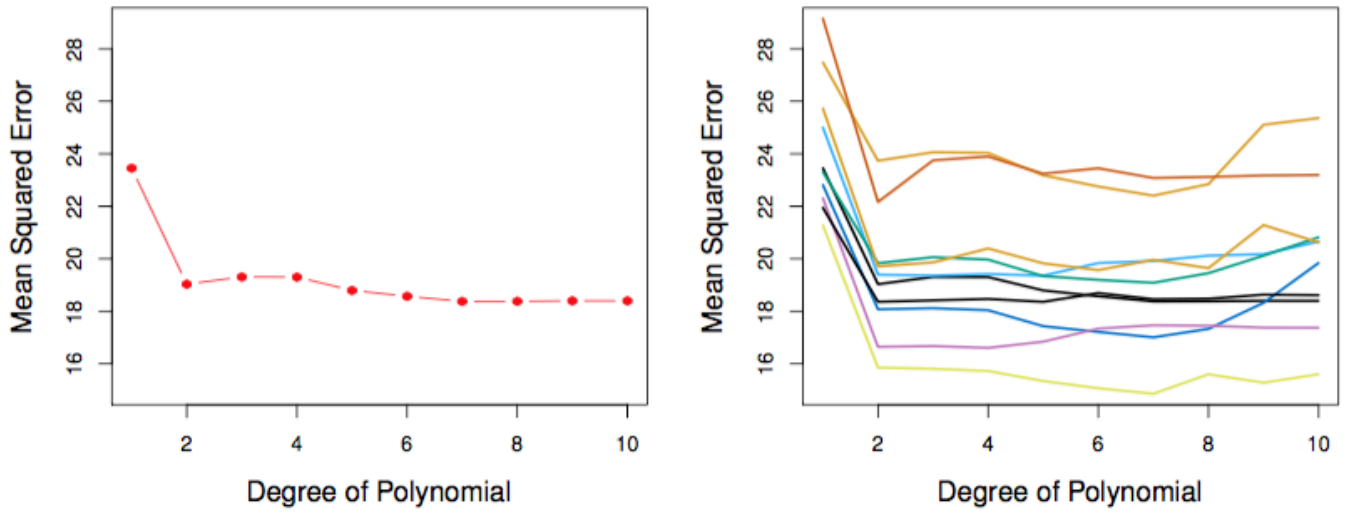
The Validation Set Approach



A random splitting into two halves: left part is training set, right part is validation set

Example

- Want to compare linear vs higher-order polynomial terms in a linear regression
- We randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.



Left panel shows single split; right panel shows multiple splits

Drawbacks of validation set approach

- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.

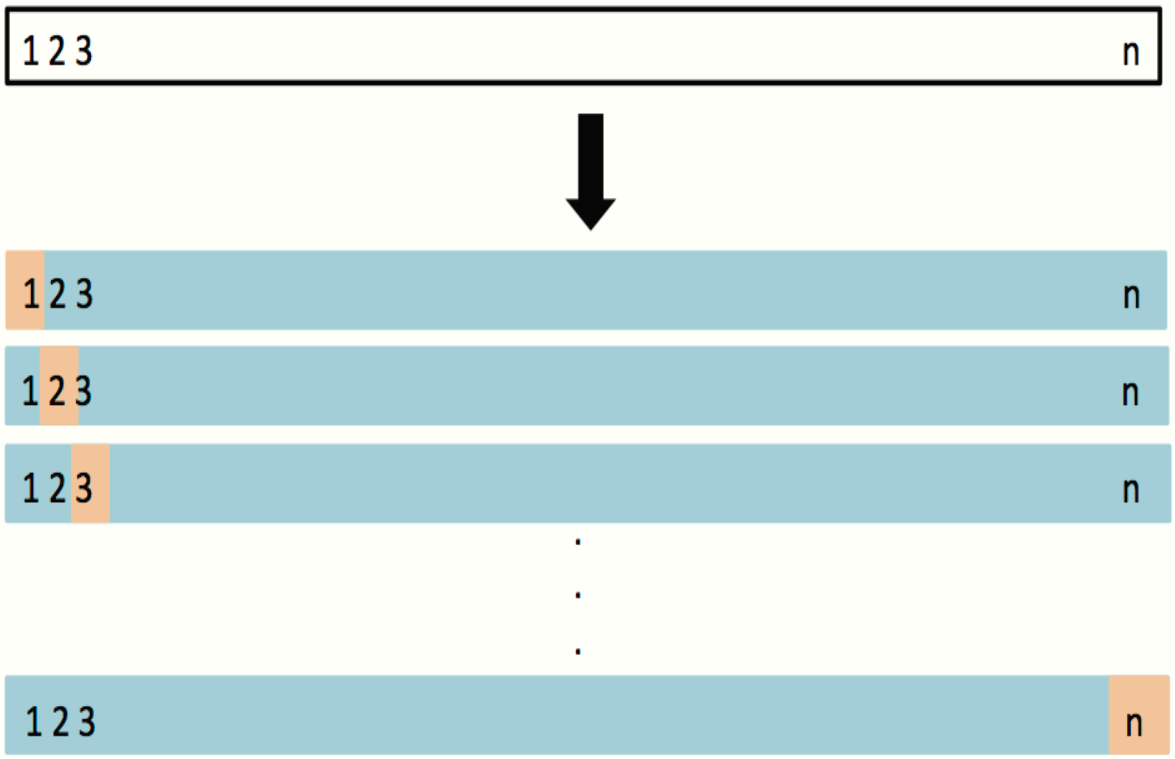
- In the validation approach, **only a subset of the observations** — those that are included in the training set rather than in the validation set — are used to fit the model.

- This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set. **WHY?**

Approaches to estimate the performance

- 3 commonly used resampling approaches
  - Validation set (shown)
  - Leave one out Cross Validation (LOOCV)
  - K-Fold Cross Validation

Leave-One-Out Cross Validation (LOOCV)

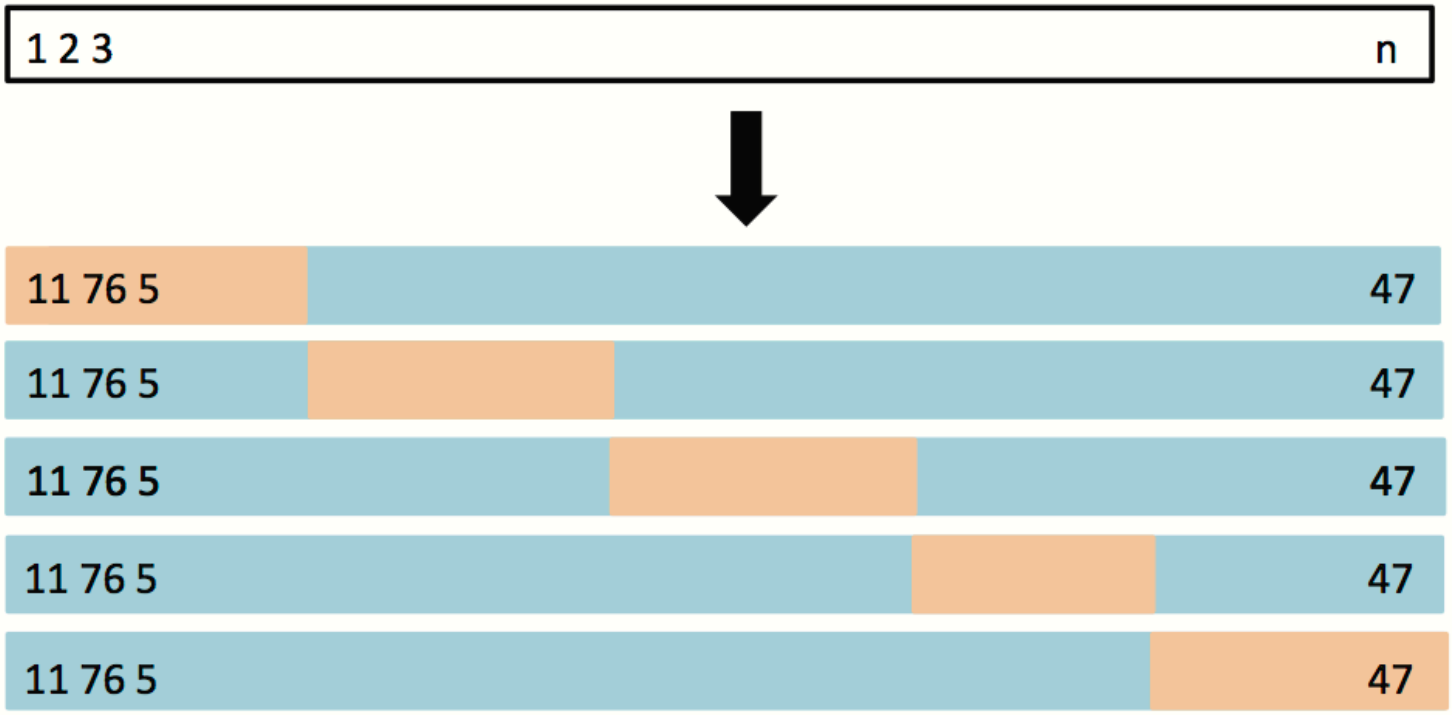


Fit w/o red sample and predict the red sample. Average over all n repeats

LOOCV vs. the Validation Set Approach

- LOOCV has less bias
  - We repeatedly fit the statistical learning method using training data that contains n-1 obs., i.e. almost all the data set is used
- LOOCV produces a less variable MSE
  - The validation approach produces different MSE when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results, because we split based on 1 obs. each time
- LOOCV is computationally intensive (disadvantage)
  - We fit the each model n times!
  - However, certain classifiers can compute LOOCV very fast
    - LDA see Aufgabe

K-fold Cross Validation



Fit w/o red samples and predict the red samples. Average over all k repeats. Do a weighted average if folds do not have the same size.

Question: What happens if k=n, what if k=2?

Details

- We divide the data into K roughly equal-sized parts $C_1, C_2, \ldots C_K$. C_k denotes the indices of the observations in part k. There are n_k observations in part k: if n is a multiple of K, then $n_k = n/K$.
- Compute

$$\mathrm{CV}_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$$

where $\mathrm{Err}_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$.

- The estimated standard deviation of CV_K is

$$\widehat{\mathrm{SE}}(\mathrm{CV}_K) = \sqrt{\sum_{k=1}^{K} (\mathrm{Err}_k - \overline{\mathrm{Err}_k})^2/(K-1)}$$

- This is a useful estimate, but strictly speaking, not quite valid. *Why not?*

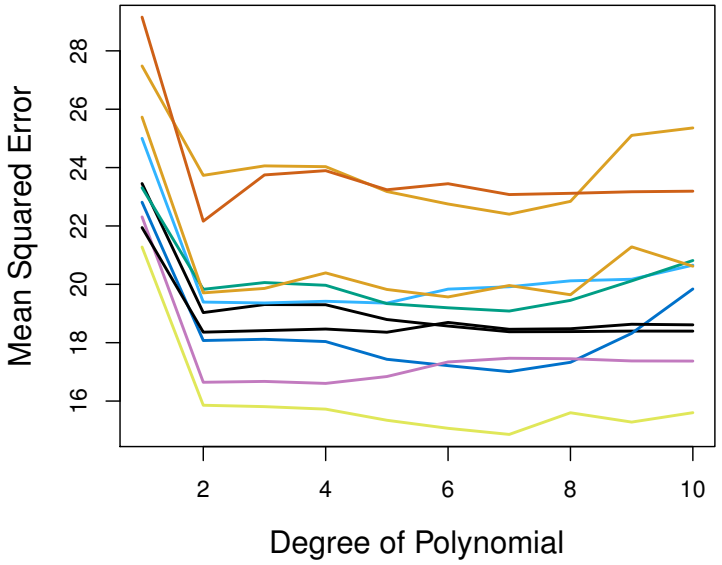
Other estimators besides Err are possible

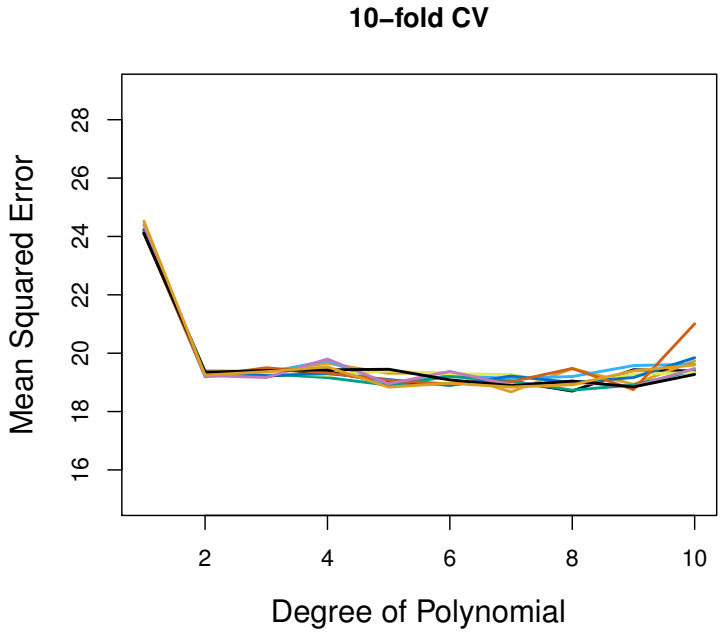
$$CV_K = \frac{1}{K} \sum_{k=1}^{K} \mathrm{Err}_k$$

All folds same size

Auto Data: Validation Set Approach vs. K-fold CV Approach

- Left: Validation Set Approach
- Right: 10-fold Cross Validation Approach
- Indeed, 10-fold CV is more stable!





K-Fold Crossvalidation in R

$$CV_K = \frac{1}{n}\sum_{k=1}^{k} n_k \mathrm{Err}_i$$

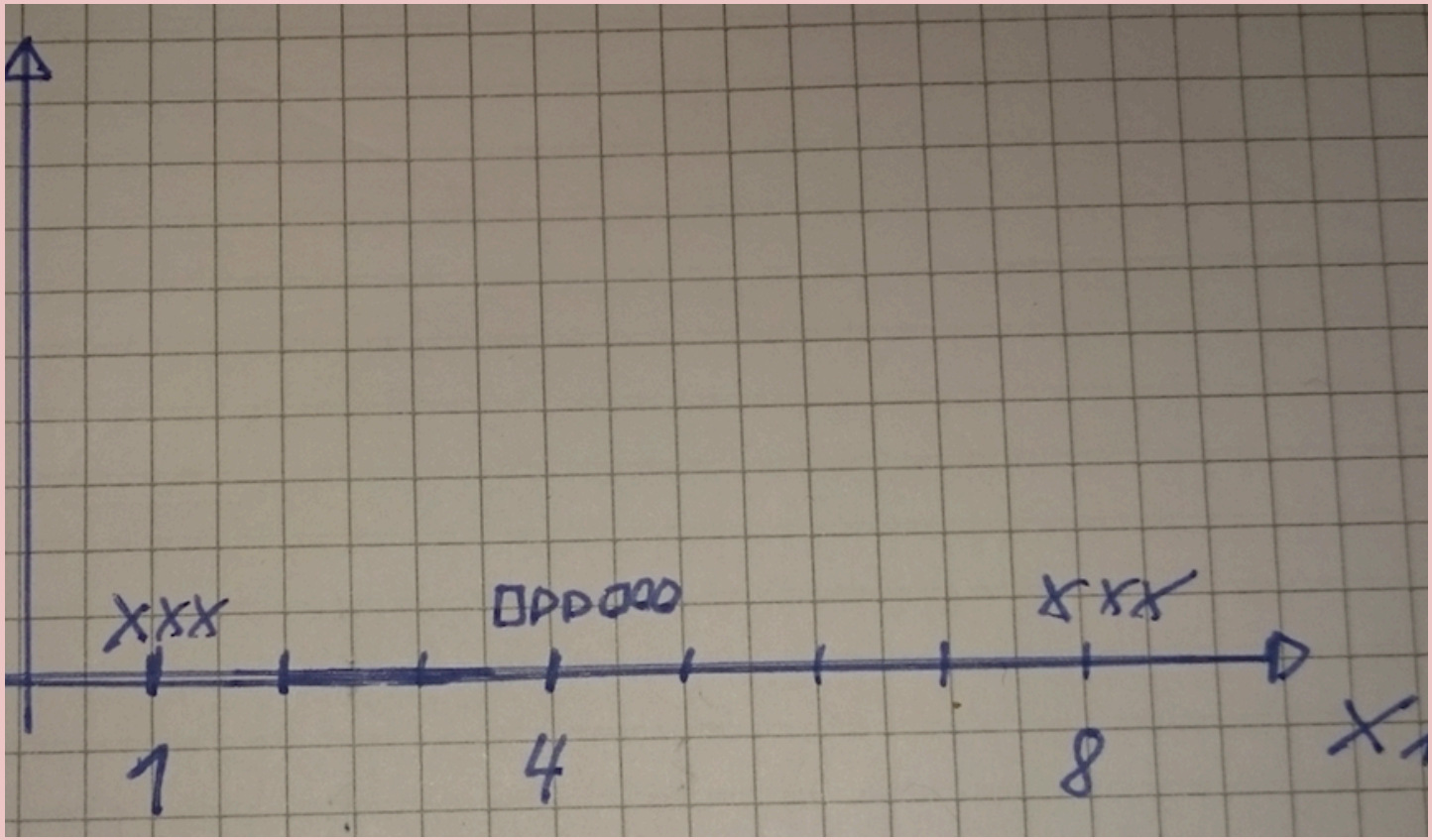
```
require(caret)
require(MASS)

test_folds = createFolds(iris$Species, 10) #Creating 10 folds
cv = 0
for (test in test_folds) {
  fit = lda(Species ~ ., data=iris[-test,])    #without the test-set
  pred_class = predict(fit, iris[test,])$class #Achtung nicht data=
  nk_Err_k = sum(pred_class != iris$Species[test])
  cv = cv + nk_Err_k
}
cv / nrow(iris)
```

Other ways of doing cross-validation are possible,
see e.g. Introduction to Statistical Learning

Example

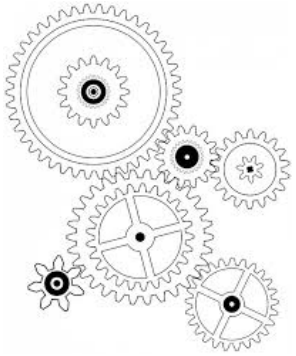
Chapter 5.1.5

Creating polynom



A. Tragen Sie eine decission boundary ein. Wie gross ist die Accuracy?
B. Führen Sie nun eine neue Variable X2 = X1^2 ein und tragen die Daten gegen X1 und X2 auf. Wie gut ist die Performance nun.

Example:



- Logistic Regression (2 features x_1, x_2)

$$z = \beta_0 + x_1\beta_1 + x_2\beta_2 = \beta^T x \quad \in [-\infty, +\infty]$$

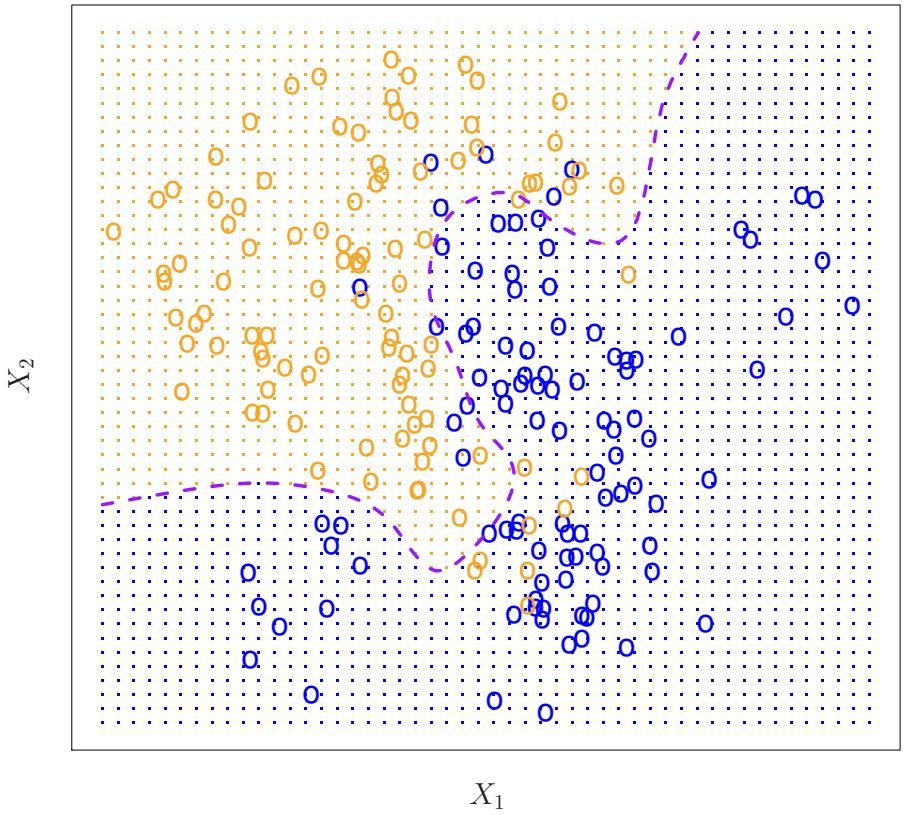
$$p_1(z) = P(Y = 1 \mid X = x) = \frac{1}{1 + e^{-z}} \quad \in [0,1]$$

- Quadratic Logistic Regression ($x \rightarrow x + x^2$)
- Higher Order ($x \rightarrow x + x^2 \ldots + x^k$)
- Simply add k columns with x^k to the data matrix

$$z = \beta_0 + x_1\beta_1 + x_2\beta_2 + x^2{}_1\beta_3 + x^2{}_2\beta_4$$

Simulated Example

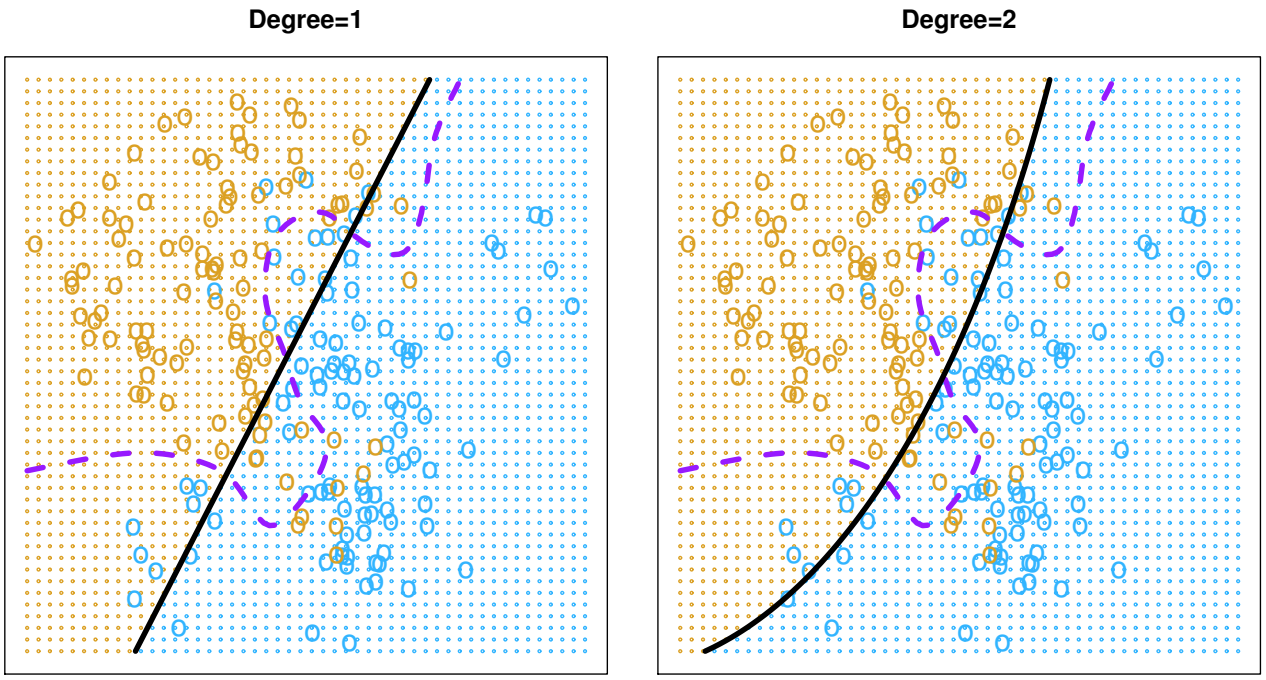
- The data set used is simulated (refer to Fig 2.13 ILSR)
- The purple dashed line is the Bayes' boundary



Bayes' Error Rate: 0.133

Simulated Example

- Linear Logistic regression (Degree 1) is not able to fit the Bayes' decision boundary
- Quadratic Logistic regression does better than linear

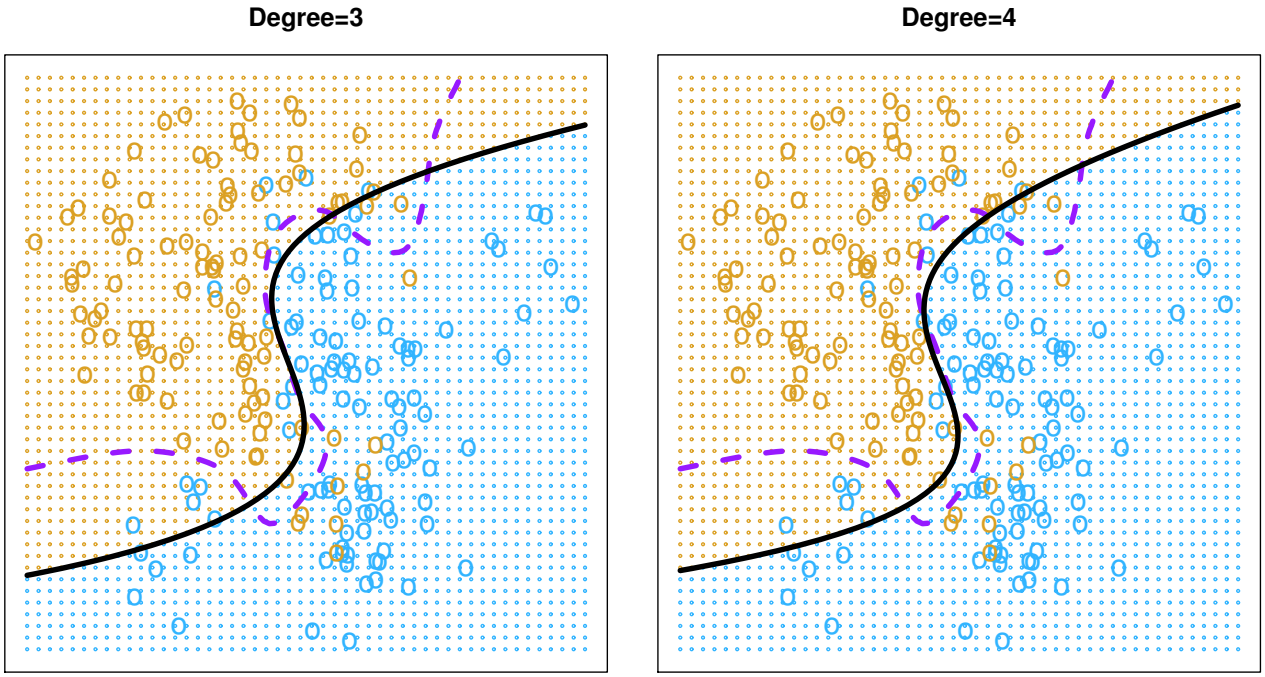


Error Rate: 0.201

Error Rate: 0.197

Simulated example

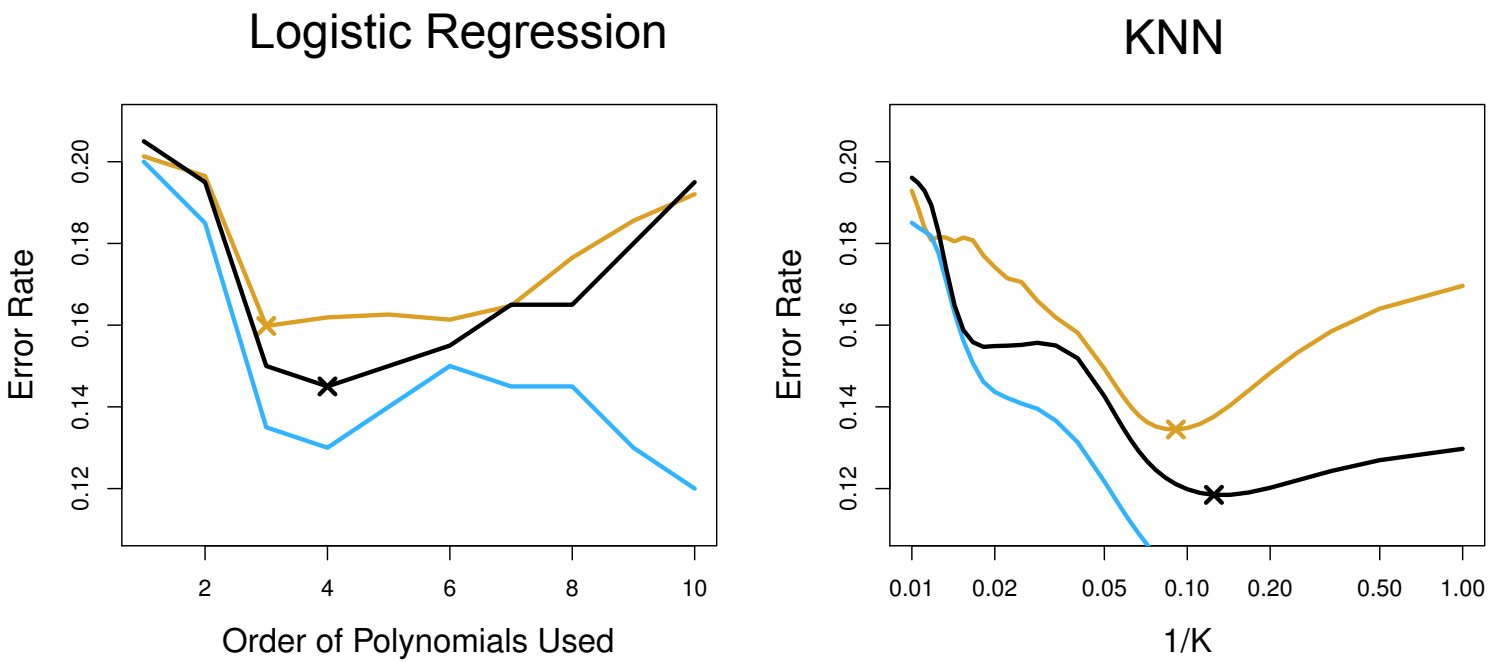
- Using cubic and quartic predictors, the accuracy of the model improves



Error Rate: 0.160

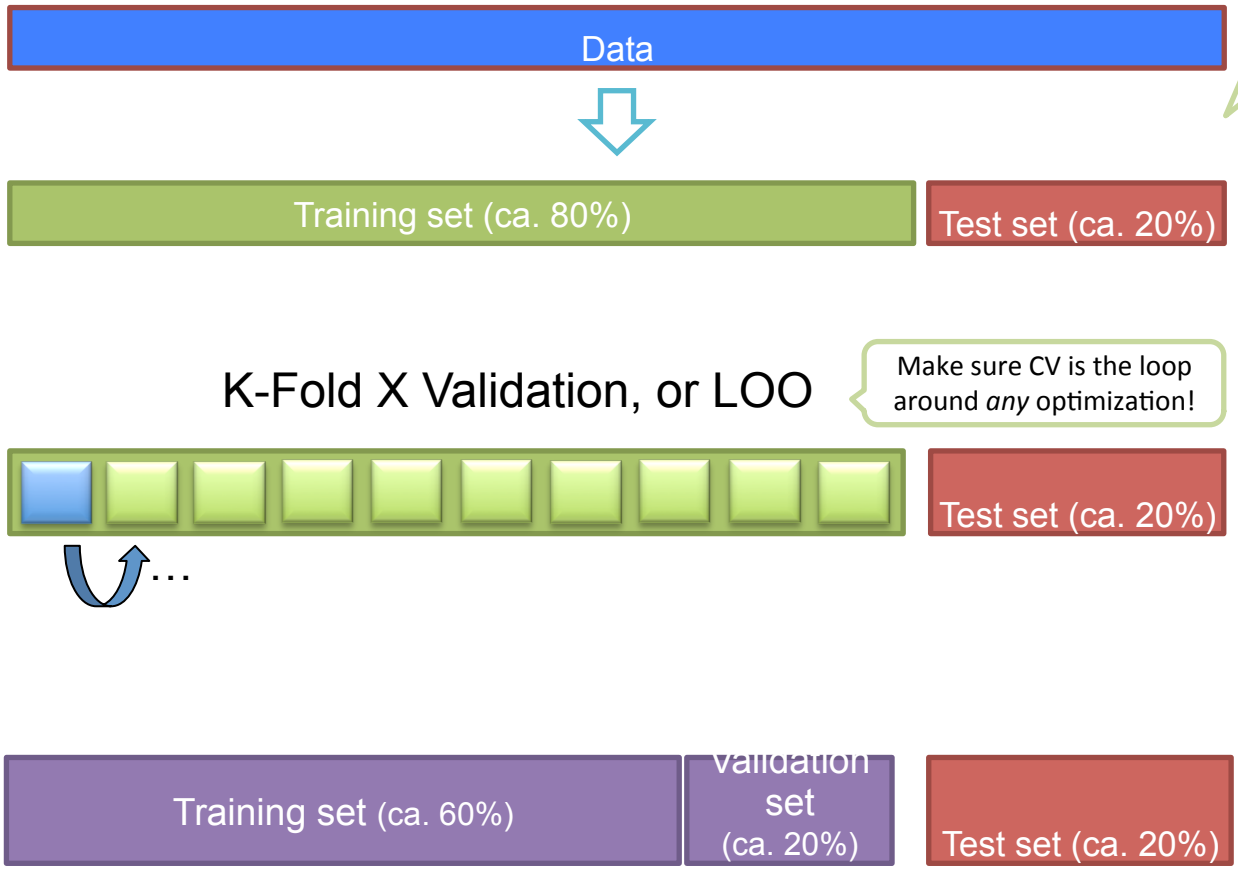
Error Rate: 0.162

CV to Choose the Order



- Brown: Test Error
- Blue: Training Error
- Black: 10-fold CV Error

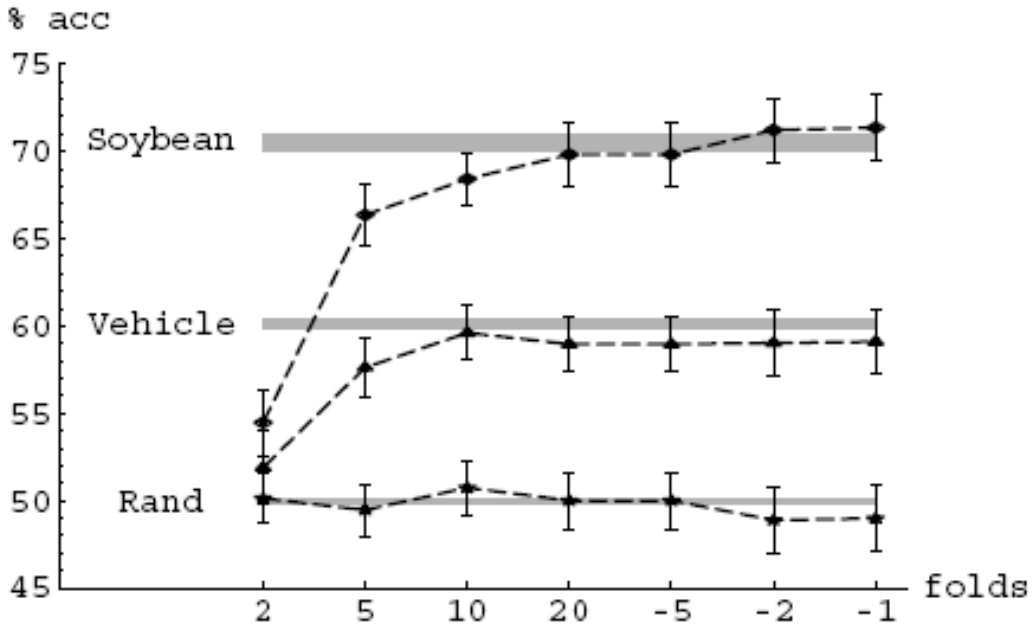
Recap / How to be on the safe side



Which LOOCV or k-Fold?

What is the best cross-validation scheme? Bias

- Example (3 Datasets)



Small training set
(sometimes larger bias)

Large training set (low bias)

- Accuracy gets better (or stays constant) the larger the training set
- Kohavi, R. (1995),A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *in* 'IJCAI', pp. 1137-1145.

What is the best cross-validation scheme? Variance

- LOOCV:
  - The data used only differs by one example (highly correlated)
  - We don't have n independent examples

$$CV = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i$$

- K-Fold:
  - The data differs by N/k examples
  - We don't have n independent examples

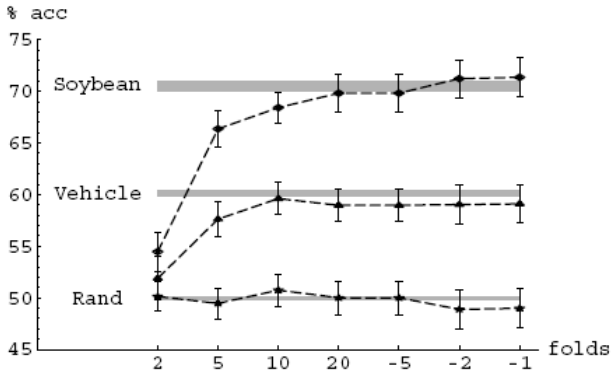
$$CV = \frac{1}{K}\sum_{i=1}^{K} \frac{n_k}{n} \mathrm{Err}_i \qquad CV = \frac{1}{K}\sum_{i=1}^{K} \mathrm{Err}_i \quad \text{if all folds have same size}$$

Question: If you repeat how much you will CV change? Observation: k-fold usually has less variance.

What is the best cross-validation scheme?

- LOOCV
  - Not random
  - Sometimes slow (there are build in procedures for some classifiers)
  - High Variance
  - Lowest possible bias (nearly the whole data seen)

- K-Fold Xvalidation
  - Random result
  - Usually faster
  - Lower Variance
  - Higher Bias (usually not problematic)



Standard today: k=10 Fold cross-validation (sometimes repeated)

More performance measures

Accuracy as performance measure

Evaluate prediction accuracy on data

Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

For an ideal classifier the off-diagonal entries should be zero: c=0, b=0, or Accuracy=1

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Simply count the # correct / all

Source: Tan, Steinbach, Kumar

Measures based on contingency tables

- Accuracy : Standard measure that doesn't regard different «costs» of errors
- Kappa statistic for inter-rater agreement: Useful to show relative improvement over random predictor

- From information retrieval domain (used far beyond!)
  - Recall : How many of the *relevant* documents
  - Precision : How many of the *returned* documents are actually *relevant*
  - F-measure : Combination of recall & precision via their harmonic mean
  - There's a **trade-off** between recall and precision because they show the two different types of error

- From medical domain
  - Sensitivity (=true positive rate, recall)
  - Specificity (=true negative rate)

- Taking all possible operating points between the two errors into account (→ see next slide)
  - AUC: Area under ROC curve
  - For recall-precision curves, the farther away from a straight line they are, the better

Slide taken from Thilo Stadelman: CAS Data Science Application

Operating on different levels of trust (motivation)

- LDA makes 252+ 23 mistakes on 10000 predictions (2.75% misclassification error rate)
- But LDA miss-predicts 252/333 = 75.5% of defaulters!
- LDA gives probability belonging to one class. Perhaps, we shouldn't use 0.5 as threshold for predicting default?

		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

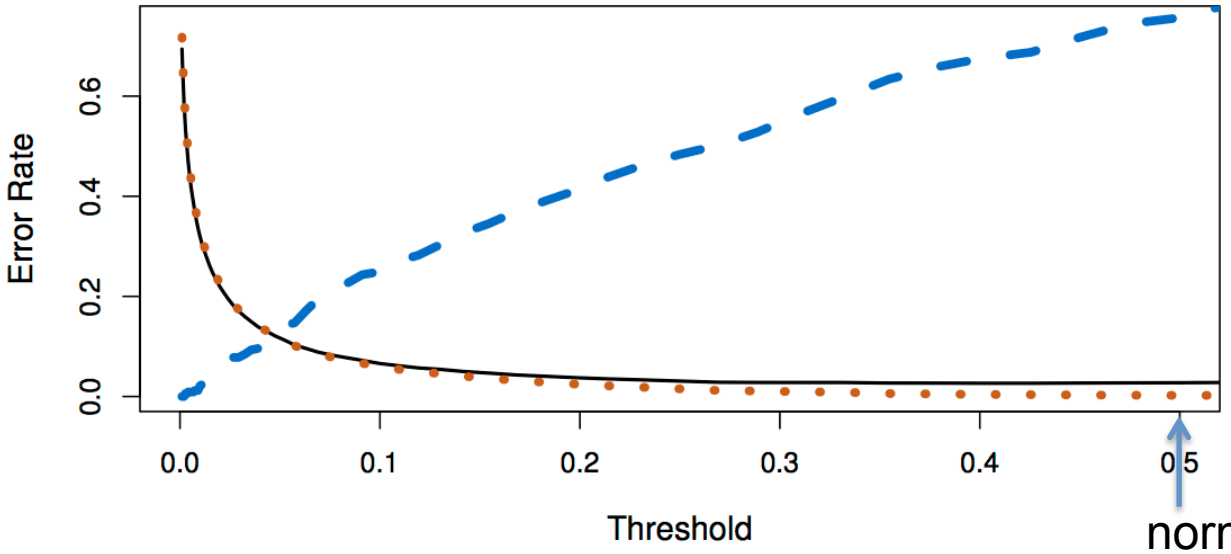
Operating on different levels of trust (motivation)

- Now the total number of mistakes is 235+138 = 373 (3.73% misclassification error rate)
- But we only miss-predicted 138/333 = 41.4% of defaulters
- We can examine the error rate with other thresholds

		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9432	138	9570
	Yes	235	195	430
	Total	9667	333	10000

Different levels in one plot

- Black solid: overall error rate
- Blue dashed: Fraction of defaulters missed
- Orange dotted: non defaulters incorrectly classified



normal operation point
LDA

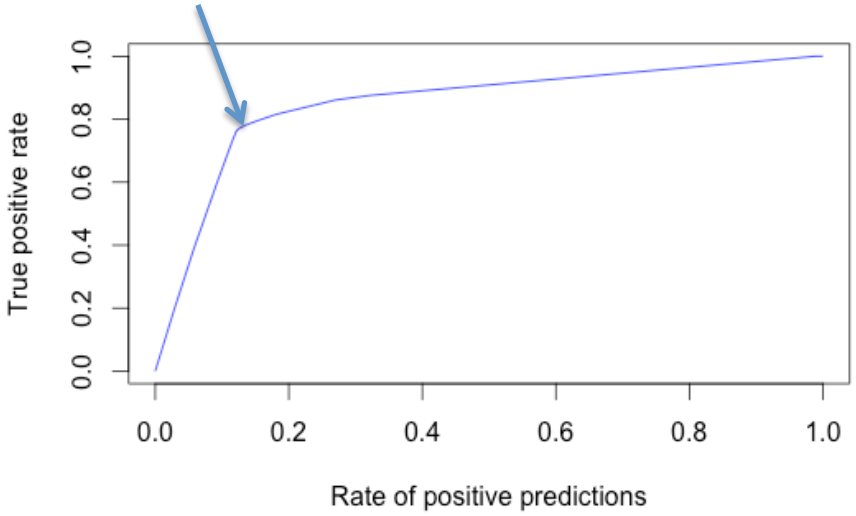
Marketing campaigns: Lift Charts (See Ruckstuhl Skript)

- Promotional mail to potential 1’000’000 households
- Only want to contact those which buy the product
- Use classifier to rank the customers

	Predicted probability	Actual class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
...	...	...

This lift chart is called cumulative lift chart

80% of all possible contacting
10% of all households



Marketing campaigns: Lift Charts

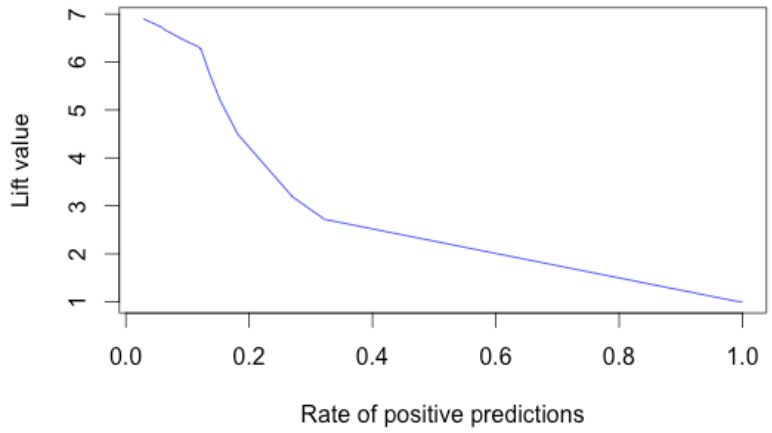
- Often Lift-Value
  - Say 1/3 are yes, 2/3 are No

	Predicted probability	Actual class
1	0.95	Yes
2	0.93	Yes
3	0.93	No
4	0.88	Yes
...	...	...

3 Yes from 4 Fraction: 3/4

By Random 1/3

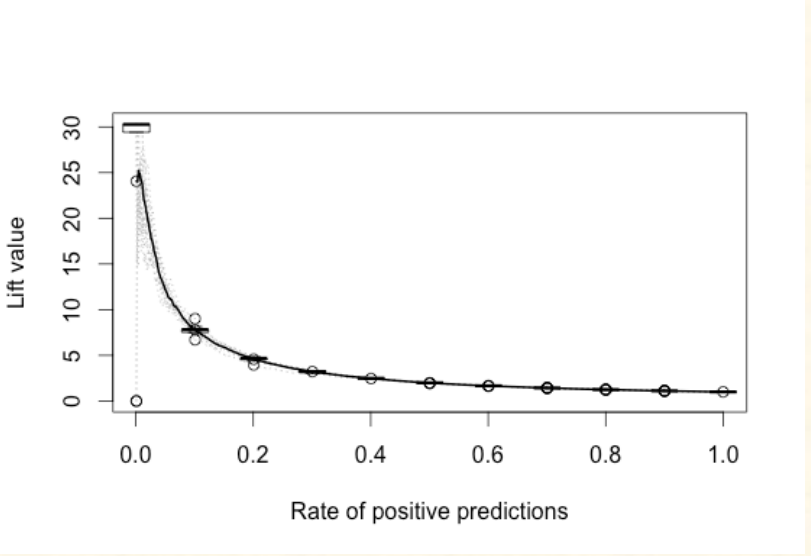
Lift = ¾ / 1/3 = 9/4



Lift Chart in R (using ROCR)


```
##############################
# ROC Curves in R (insample)
library(ROCR)
library(ISLR)
fit = lda(default ~ ., data=Default)
preds = predict(fit, Default)$posterior
head(preds)
pred <- prediction(preds[,2], Default$default)
perf <- performance(pred,measure="lift", x.measure="rpp")
plot(perf, main="lift curve")

# Cumulativ, fraction of those who responed
perf <- performance(pred,measure="tpr", x.measure="rpp")
plot(perf, main="lift curve cumulative")
```

Lift Chart in R (using ROCR, k-Fold)




```
############################
# Lift Charts for k-Fold
preds = list()
labels = list()
test_folds = createFolds(Default$default, 10)
i = 1
for (test in test_folds) {
  fit = lda(default ~ ., data=Default[-test,])
  preds[[i]] = predict(fit, Default[test,])$posterior[,2]
  labels[[i]] = Default[test,]$default
  i = i + 1
}
pred <- prediction(preds, labels)
perf <- performance(pred,"lift","rpp")
performance(pred,"auc")
plot(perf,col="grey",lty=3)
plot(perf,lwd=2,avg="vertical",spread.estimate="boxplot",add=TRUE)
```

Ende der zweiten Stunde 2015
Nicht dabei ROCR Kurven

ROC curves

ROC curves

- **R**eceiver **o**peration **c**haracteristics used in WWII as a trade-off between recall and precision for radar systems. To correctly detected Japanese aircraft from their radar signals (after Pearl Harbor).
- Like radar systems classifiers can be “operated” on different levels of trust. E.g. only take those to a class which have a very high affinity to that class.

		actual value: *p*	actual value: *n*	total
prediction outcome	*p'*	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
total		P	N	

True Positive Rate

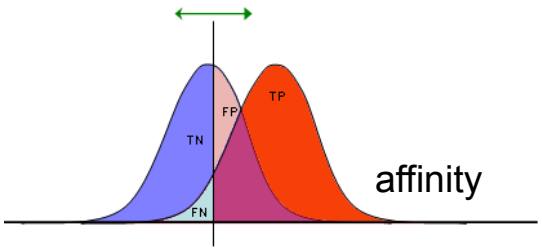
TPR = TP / P = TP / (TP + FN)

False Positive Rate

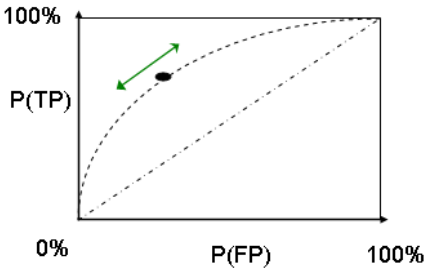
FPR = FP / N = FP / (FP + TN)

Accuracy

ACC = (TP + TN) / (P + N)



TP	FP
FN	TN
1	1



taken from Wikipedia

Einzeichnen von Hand

		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

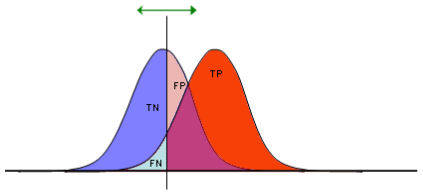
		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9432	138	9570
	Yes	235	195	430
	Total	9667	333	10000

True Positive Rate

TPR = TP / P = TP / (TP + FN)

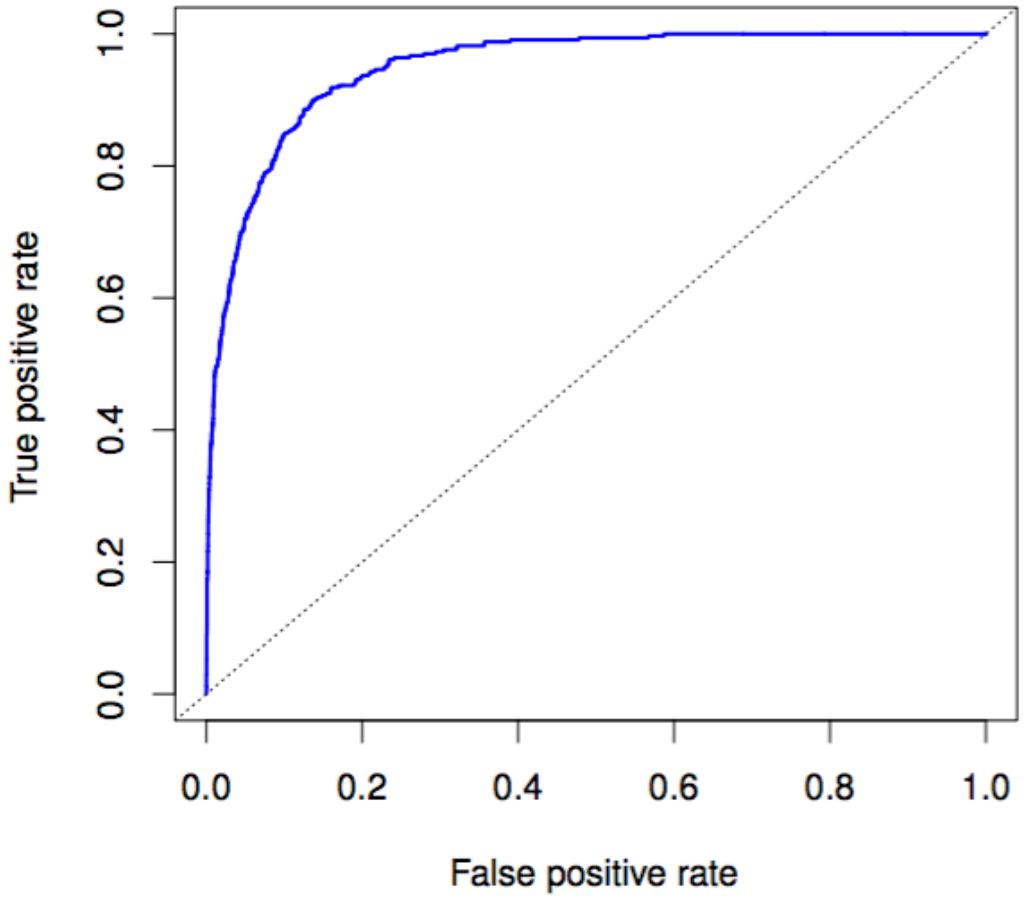
False Positive Rate

FPR = FP / N = FP / (FP + TN)



TP	FP
FN	TN
1	1

ROC-Curve for the example



Area under Curve AUC is often used. AUC = 1 perfect, AUC=0.5 random.

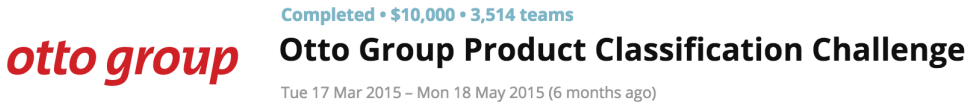
ROC-Curve in R (using ROCR)


```
##############################
# ROC Curves in R (insample)
library(ROCR)
library(ISLR)
fit = lda(default ~ ., data=Default)
preds = predict(fit, Default)$posterior
head(preds)
pred <- prediction(preds[,2], Default$default)
plot(performance(pred,"tpr","fpr"))
performance(pred, "auc")
```

Praktikum

Bewertete Hausaufgabe

- Mitmachen an einer Data Science Challenge
- Erste Möglichkeit [Otto Produkt Klassifikation]



- Einreichen unter:
  - http://srv-lab-t-864/submission/Otto/
- Leaderboard:
  - http://srv-lab-t-864/leaderboard/Otto/
- Andere Challenges von Kaggle
  - Nach Rücksprache können Sie auch an einer anderen Kaggle Challenge teilnehmen (nicht Titanic)
  - Zum Beispiel: MNIST
  - Beachten Sie, es muss ein Klassifizierungsproblem sein.
  - Username muss dann mitgeteilt werden

Bewertete Hausaufgabe

- 2-3er Teams OK
- Teams melden bis 9 Dezember
- Vorstellung im letzten Praktikum (16.12.2015)
  - Etwa 10-20 Minuten
- Einreichen der Lösung
- Bewertung in halben Noten
  - Performance
  - Vortrag
  - Folien
- Note zählt nur zur Verbesserung!

Example

```
setwd("~/Dropbox/__ZHAW/StDM/Vorbereitung_HS2015/2015_HS/challenge/Otto")
X_Train = read.table("train_otto.csv", sep=';', header = TRUE, stringsAsFactors = FALSE)
X_Test = read.table("test_otto.csv", sep=';', header = TRUE, stringsAsFactors = FALSE)

# LDA
library(MASS)
fit = lda(target ~ ., data = X_Train)
res = predict(fit, X_Test)
df = data.frame(key=X_Test$id, value=res$class)
write.table(x=df, file = 'predictions_lda.csv', sep=';', row.names = FALSE)
```