KAGGLE WORKSHOP

JAN-HENDRIK RUETTINGER

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

LIKE

WHAT IS KAGGLE?

- founded in 2010 (bought by Google in 2017)
- Platform for Data Science Competitions
- +550.000 registered users
- +3500 submissions per day

kaggle™

HOW DOES A KAGGLE COMPETITION WORK?

Demo on website

AGENDA I

- Machine Learning introduction
- Linear models
- Pandas introduction
- Exploratory Data Analysis (EDA)
- Feature engineering
- Model evaluation and cross validation
- Regularization
- Decision Trees
- K-nearest Neighbor

AGENDA II

- Hyperparameter optimization
- Ensemble methods
- Lunch break
- Introduction team challenge
- Time to work on the challenge
- Short presentation of the two best solutions
- Experts on kaggle
- LIKE + Kaggle = ❤ ?

WHAT IS THE GOAL OF MACHINE LEARNING?

Target

Feature

	PassengerId	Survived	Pclass	Name	Sex	Age
0	1	0	3	Braund, Mr. Owen Harris	male	22.0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	5	0	3	Allen, Mr. William Henry	male	35.0

Feature #1

Feature #2 → $\boxed{\text{Model}}$ → Target

…

REGRESSION AND CLASSIFICATION

Feature #1

Feature #2 → Model → Target

…

Target

Class

- survived/not survived
- dog/cat

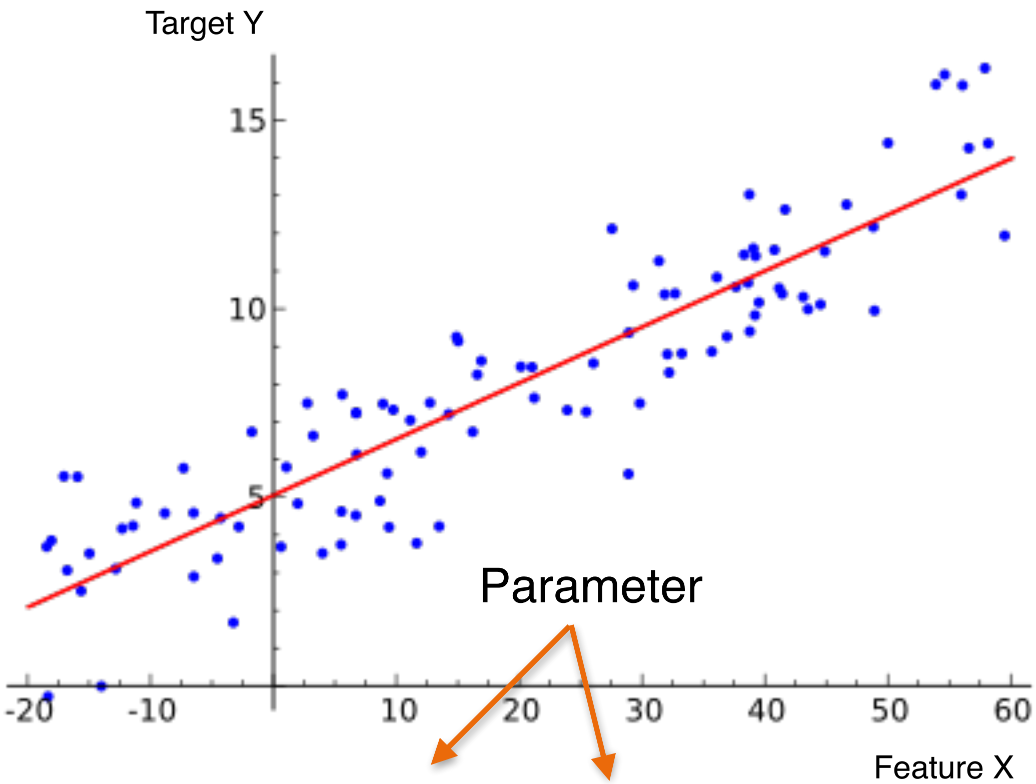
Classification

Cont. Value

- 1023
- 17.562

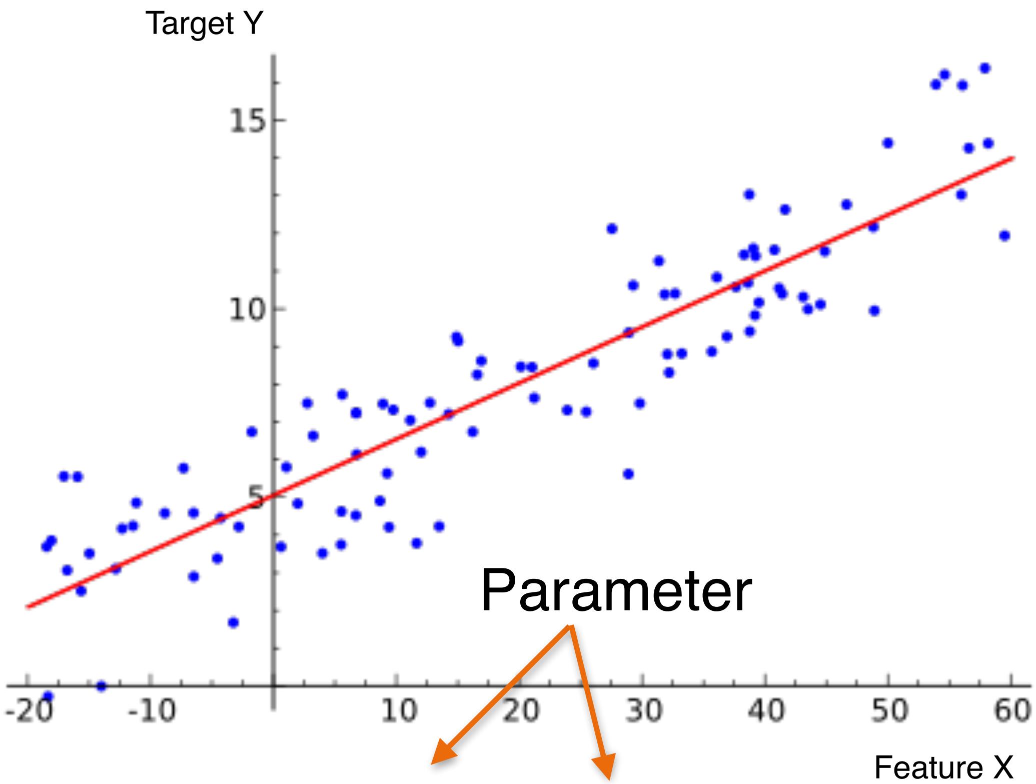
Regression

LINEAR MODELS (REGRESSION)



$$h_\theta(x) = \theta_0 + \theta_1 x$$

LINEAR MODELS (REGRESSION)



$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$$

HOW DO WE FIND THE OPTIMAL PARAMETERS?

=> Minimize a suitable cost function

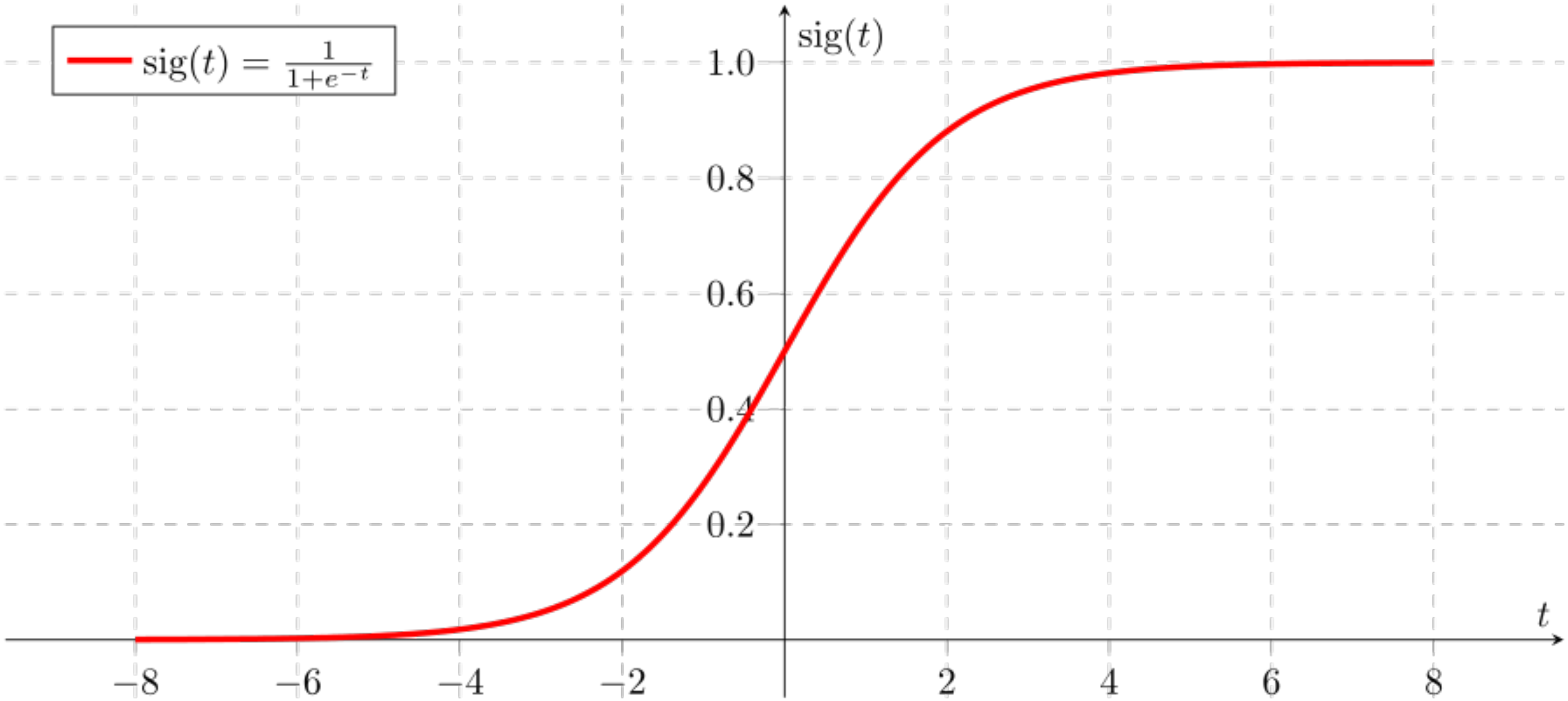
target

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{\text{Daten}} |h_\theta(x^{(i)}) - y^{(i)}|$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{\text{Daten}} (h_\theta(x^{(i)}) - y^{(i)})^2$$

- Only binary classification for now
- Sigmoid function + linear regression



FROM LINEAR REGRESSION TO LINEAR CLASSIFICATION I

Hypothesis:

$$h_{\theta,classification}(x) = sig(h_\theta(x))$$

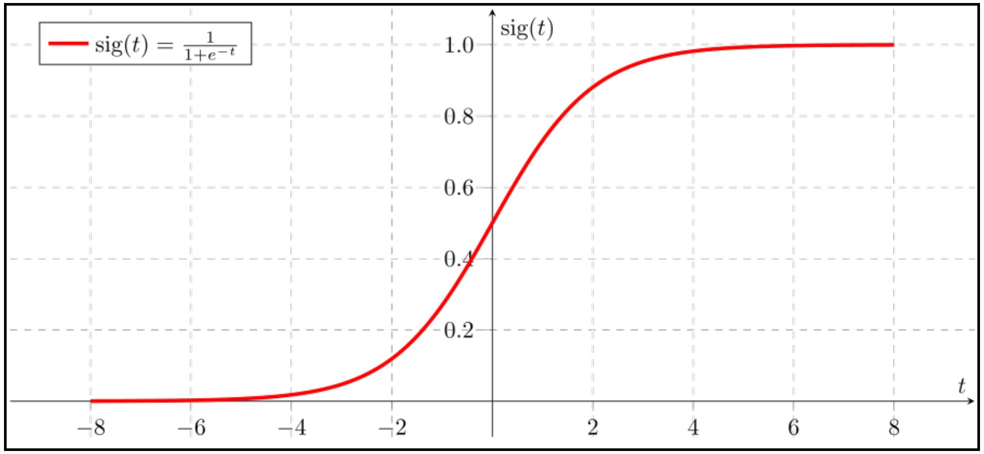
$$h_{\theta,classification}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

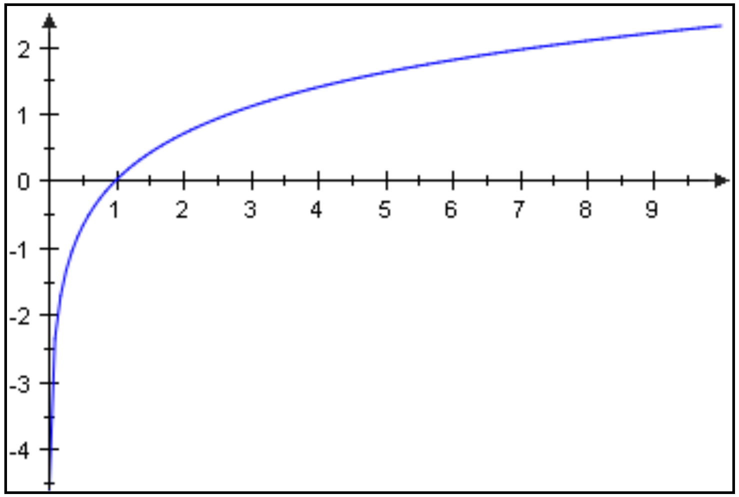
Kostenfunktion:

$$J(\theta) = -\frac{1}{m}[\sum_{\text{Daten}} y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$

INTUITION BEHIND THE COST FUNCTION

$$J(\theta) = -\frac{1}{m}[\sum_{\text{Daten}} y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$



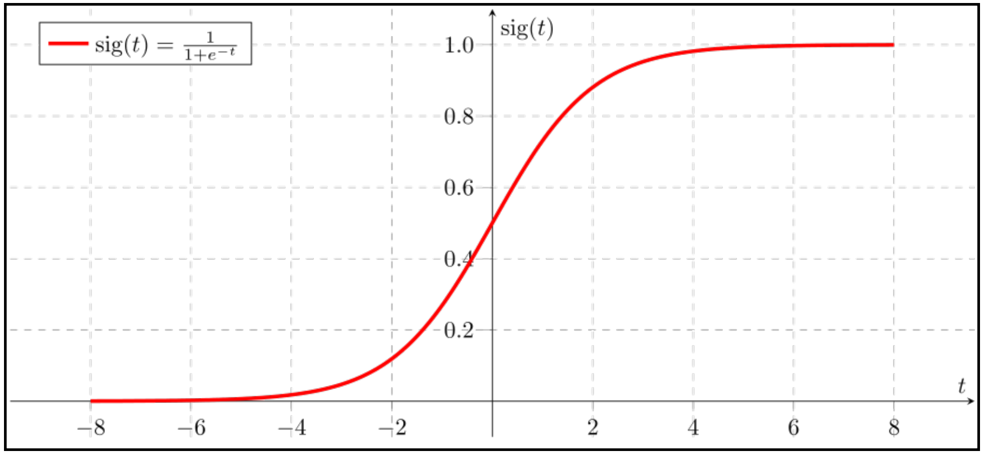


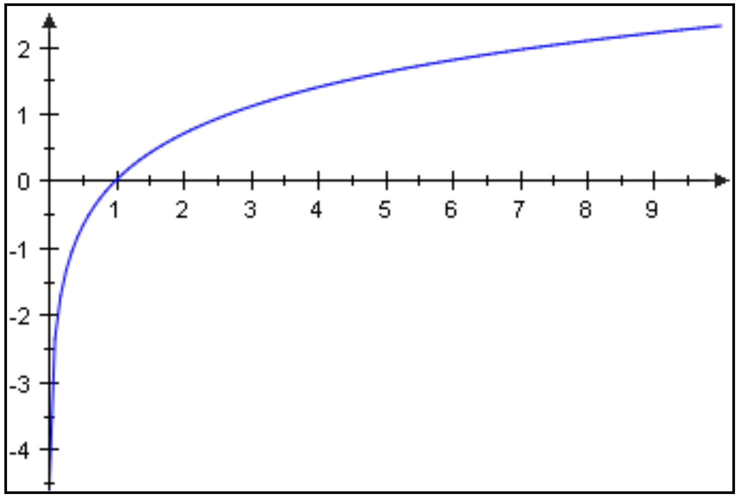
$$y = 1 \text{ und } h_\theta(x) \approx 1$$

$$J(\theta) = ?$$

INTUITION BEHIND THE COST FUNCTION

$$J(\theta) = -\frac{1}{m}[\sum_{\text{Daten}} y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$



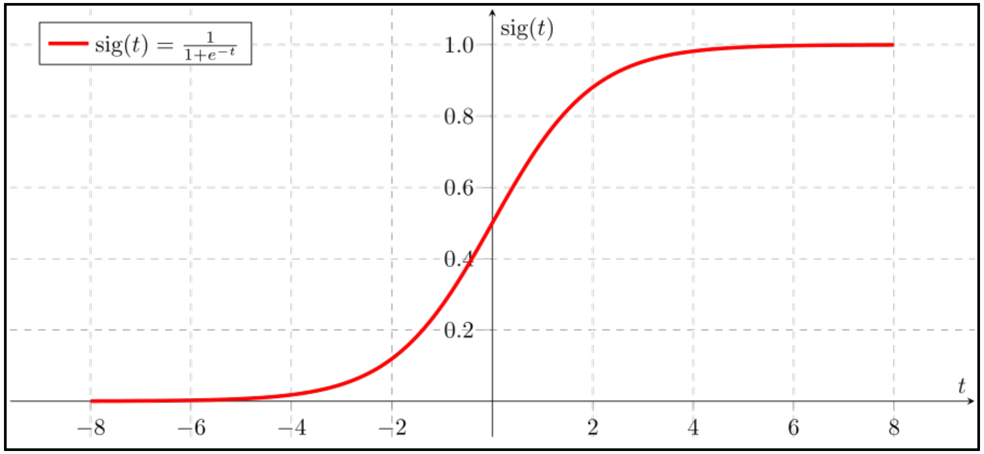


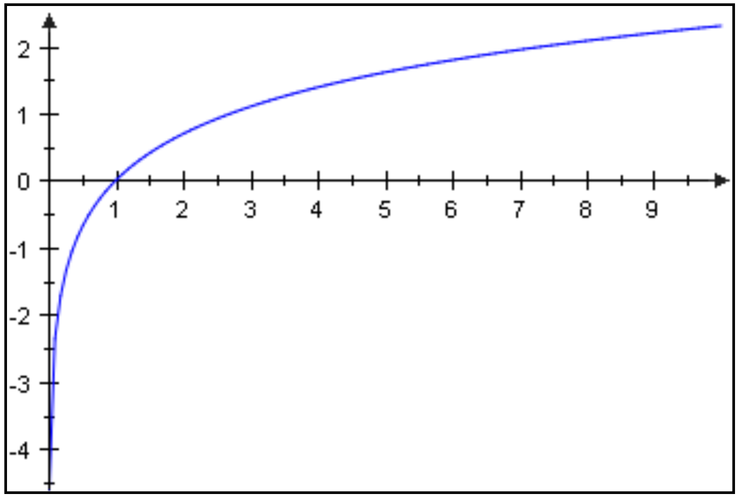
$$y = 1 \text{ und } h_\theta(x) \approx 1$$

$$J(\theta) = 0$$

INTUITION BEHIND THE COST FUNCTION

$$J(\theta) = -\frac{1}{m}[\sum_{\text{Daten}} y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$



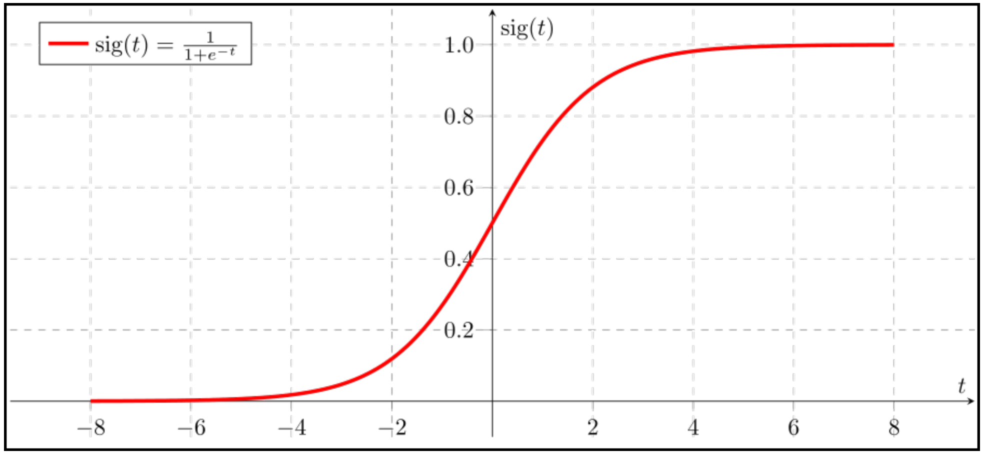


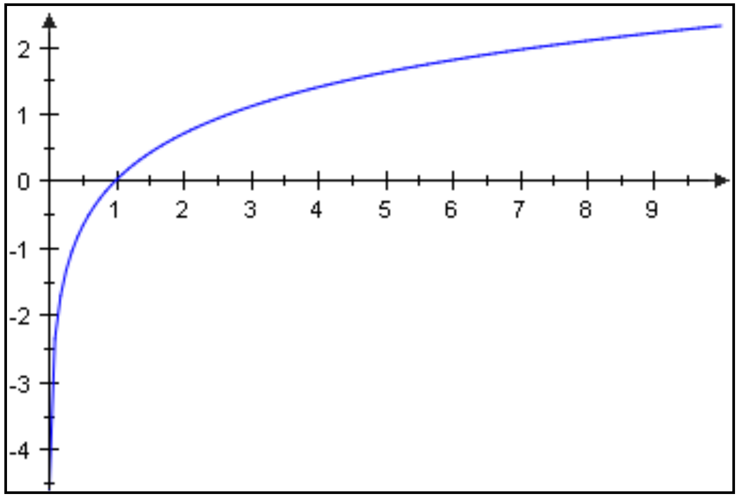
$$y = 0 \text{ und } h_\theta(x) \approx 1$$

$$J(\theta) =?$$

INTUITION BEHIND THE COST FUNCTION

$$J(\theta) = -\frac{1}{m}[\sum_{\text{Daten}} y^{(i)} log(h_\theta(x^{(i)})) + (1-y^{(i)}) log(1-h_\theta(x^{(i)}))]$$

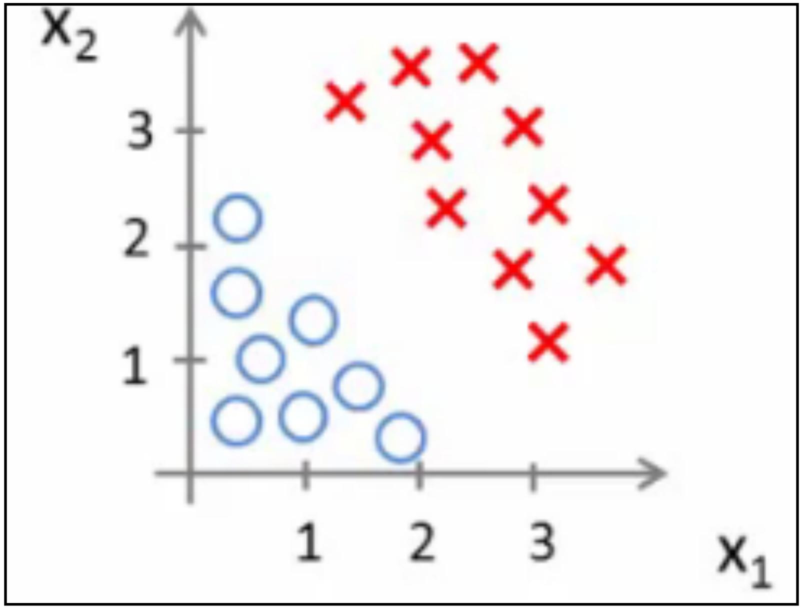


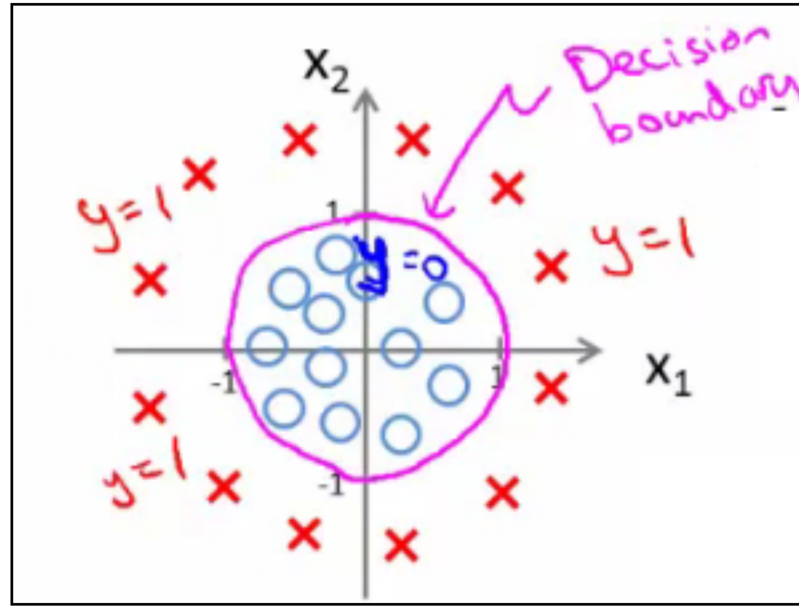


$$y = 0 \text{ und } h_\theta(x) \approx 1$$

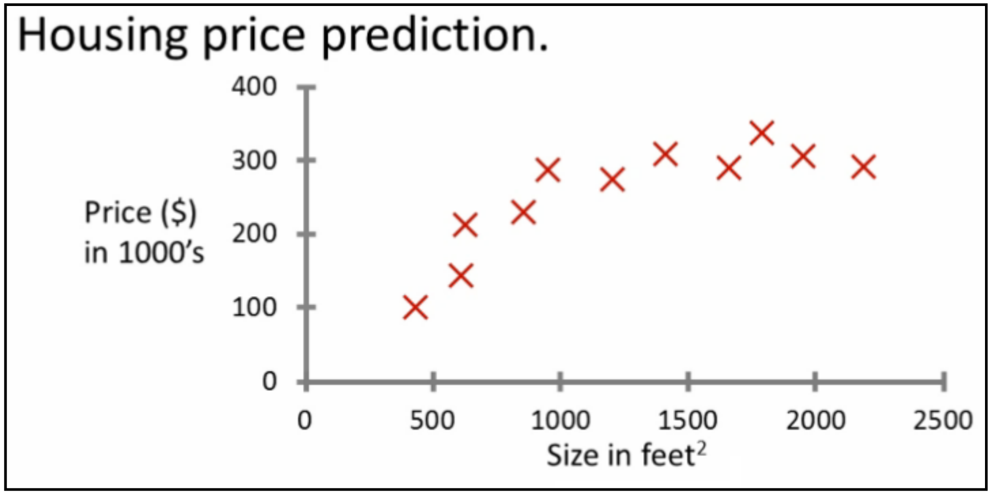
$$J(\theta) \to \infty$$

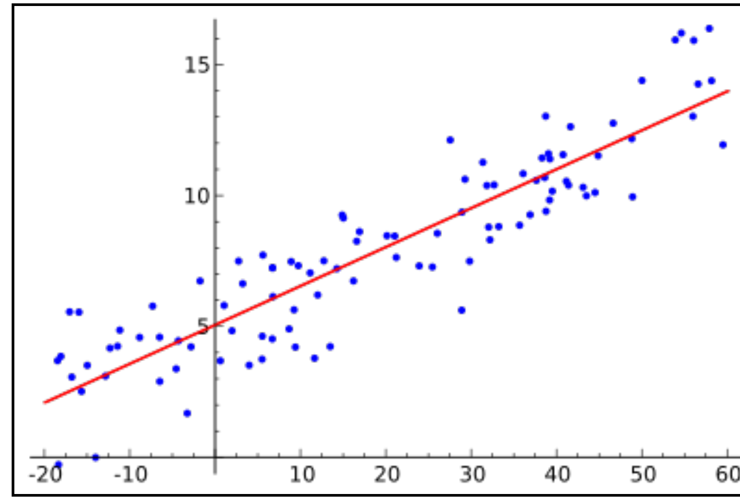
LIMITATIONS OF LINEARE MODELS (CLASSIFICATION)





LIMITATIONS OF LINEAR MODELS (REGRESSION)





VORGEHENSWEISE IM WETTBEWERB

1. Treat missing data
2. EDA (Exploratory Data Analysis)
=> Get a feeling for the data
3. Feature Engineering
=> Create/delete/transform/select

80% der Zeit

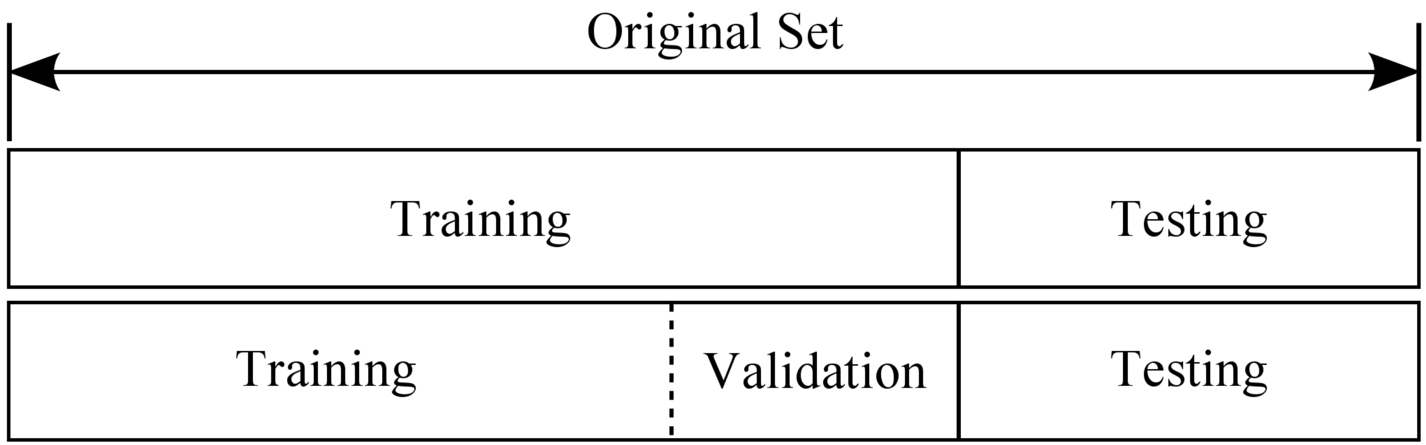
4. Build a validation scheme
=> balanced/imbalanced target?
5. Train your model

20% der Zeit

JUPYTER NOTEBOOKS

- 00_Pandas_Basics
- 01_Titanic_EDA
- 02_Data_Cleaning
- 03_Feature_Engineering
- 04_Models (Linear models)

MODEL EVALUATION AND CROSS VALIDATION



1. Fit model to training data
2. Evaluate model with validation data
3. Improve model

example exam preparation

4. Test model with test data

EXAMPLE: EXAM PREPARATION

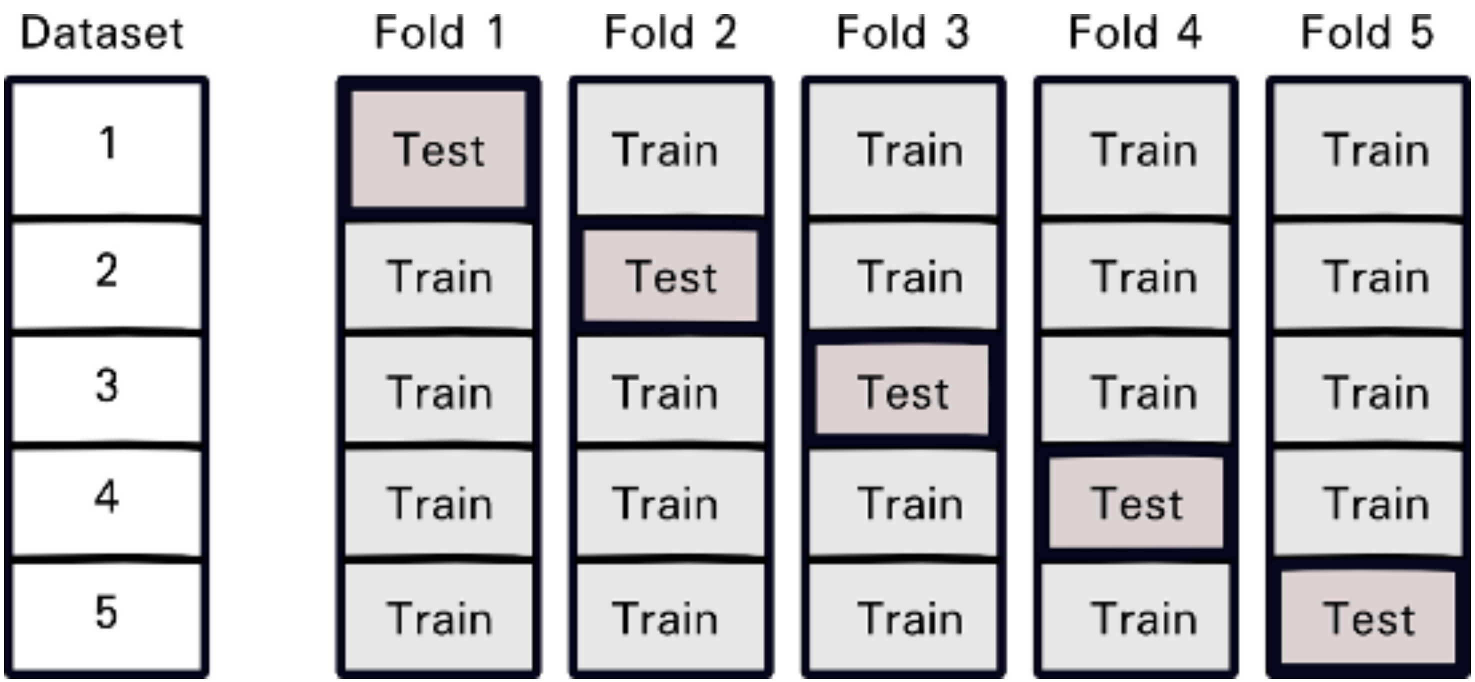
1. Study time (= model fitting)
2. Test exams (= model evaluation)
3. Revise some topics (= model improvement)

4. Real exam (= final test)

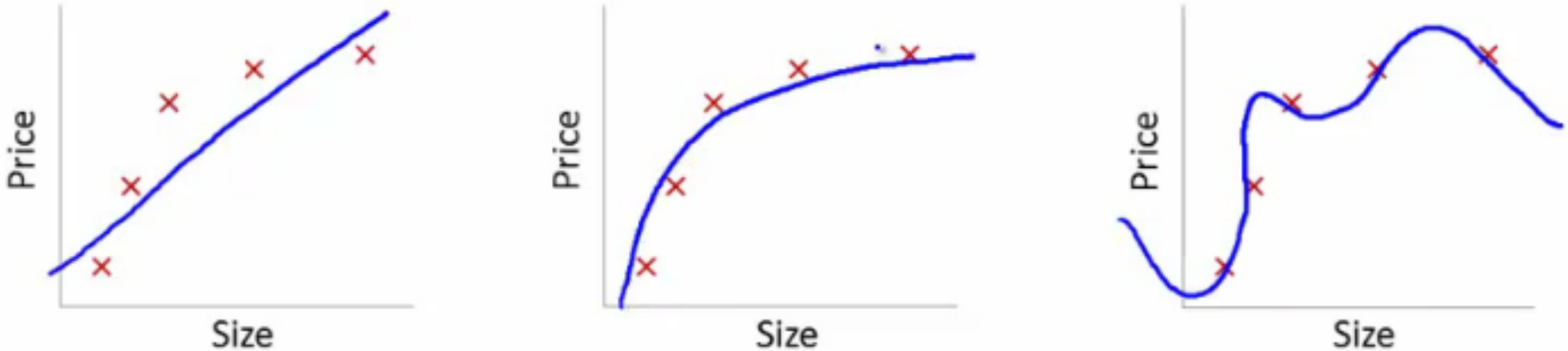
Question: What happens to your final score when your test exams are from 20 years ago?

K-FOLD VALIDATION

K-Fold validation

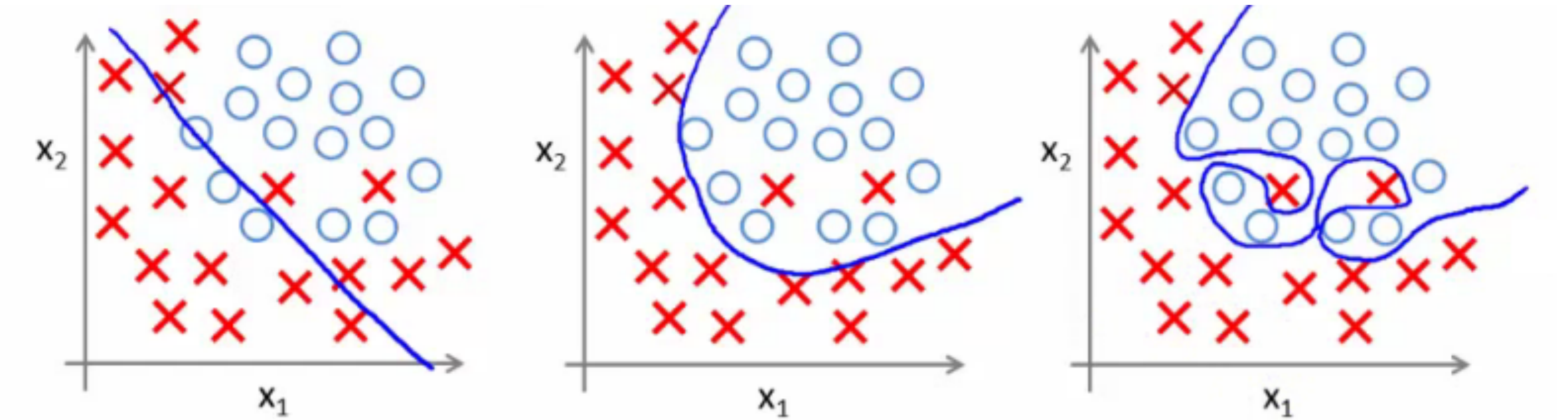


OVERFITTING VS UNDERFITTING



$\theta_0 + \theta_1 x$	$\theta_0 + \theta_1 x + \theta_2 x^2$	$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
High bias (underfit)	"Just right"	High variance (overfit)

OVERFITTING VS UNDERFITTING



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

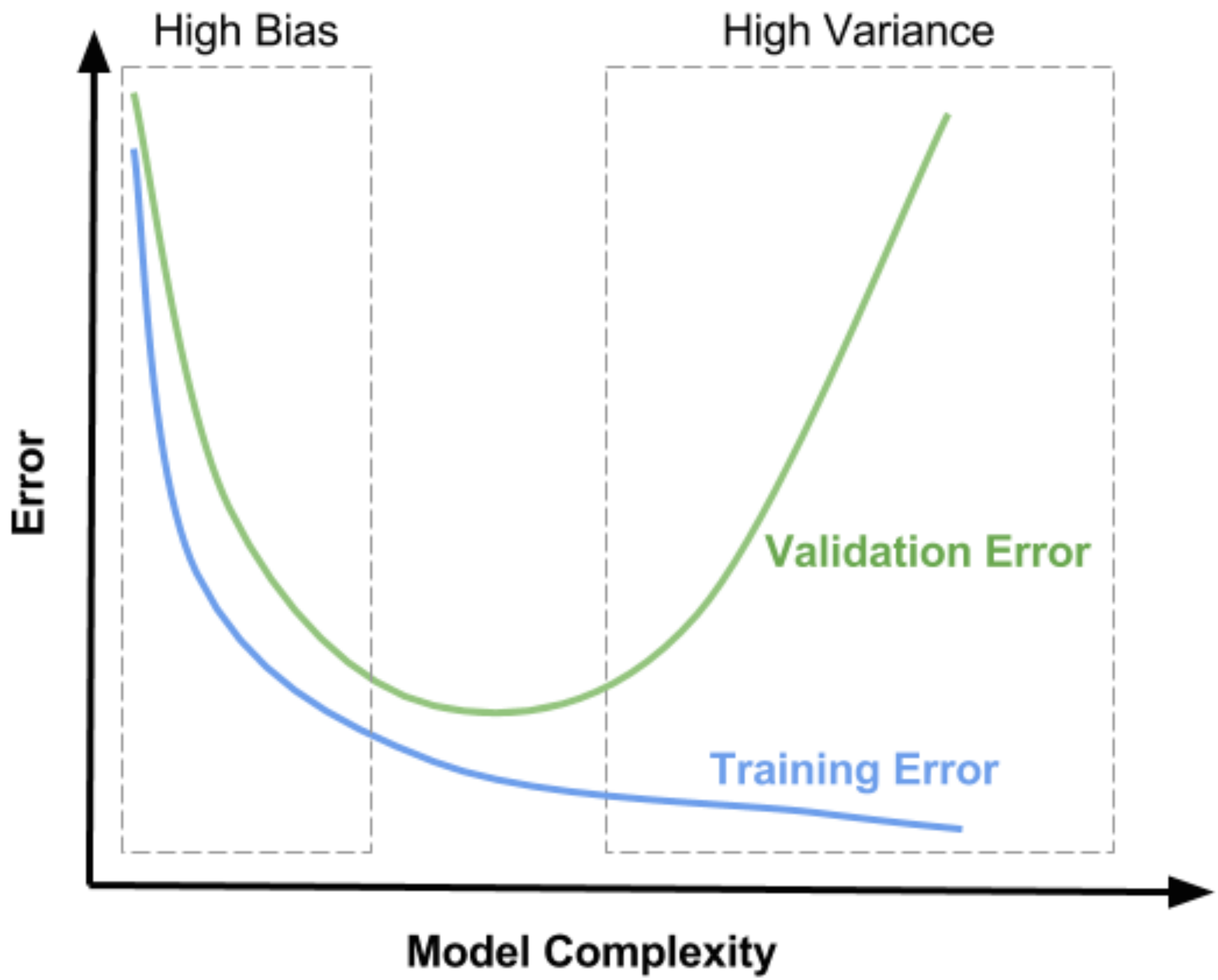
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots$$

UNDERFITTING
(high bias)

OVERFITTING
(high variance)

VALIDATION ERROR VS TRAINING ERROR



OVERFITTING VS UNDERFITTING

Overfitting	Underfitting
Fails to generalize	Fails to generalize
More training data	Increase number of features
Reduce number of features	
Regularization	

- Prevents model from overfitting
- Adds additional term/noise to cost function
- For linear models:

Hyperparameter

L1:

$$J(\theta_0, \theta_1) = \frac{1}{2m}[\sum_{m}^{i}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{n}^{j} |\theta_j|]$$

L2:

$$J(\theta_0, \theta_1) = \frac{1}{2m}[\sum_{m}^{i}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{n}^{j} \theta_j^2]$$

REGULARIZATION

L1: sklearn.linear_models.Lasso

L2: sklearn.linear_models.Ridge

=> Demo

=> First submission

=> Error analysis

DECISION TREE

- Classification and regression
- Non-linear model
- Easy to interpret
- Handles missing data well
- Performs well with large data sets
- NP hard to find optimal tree

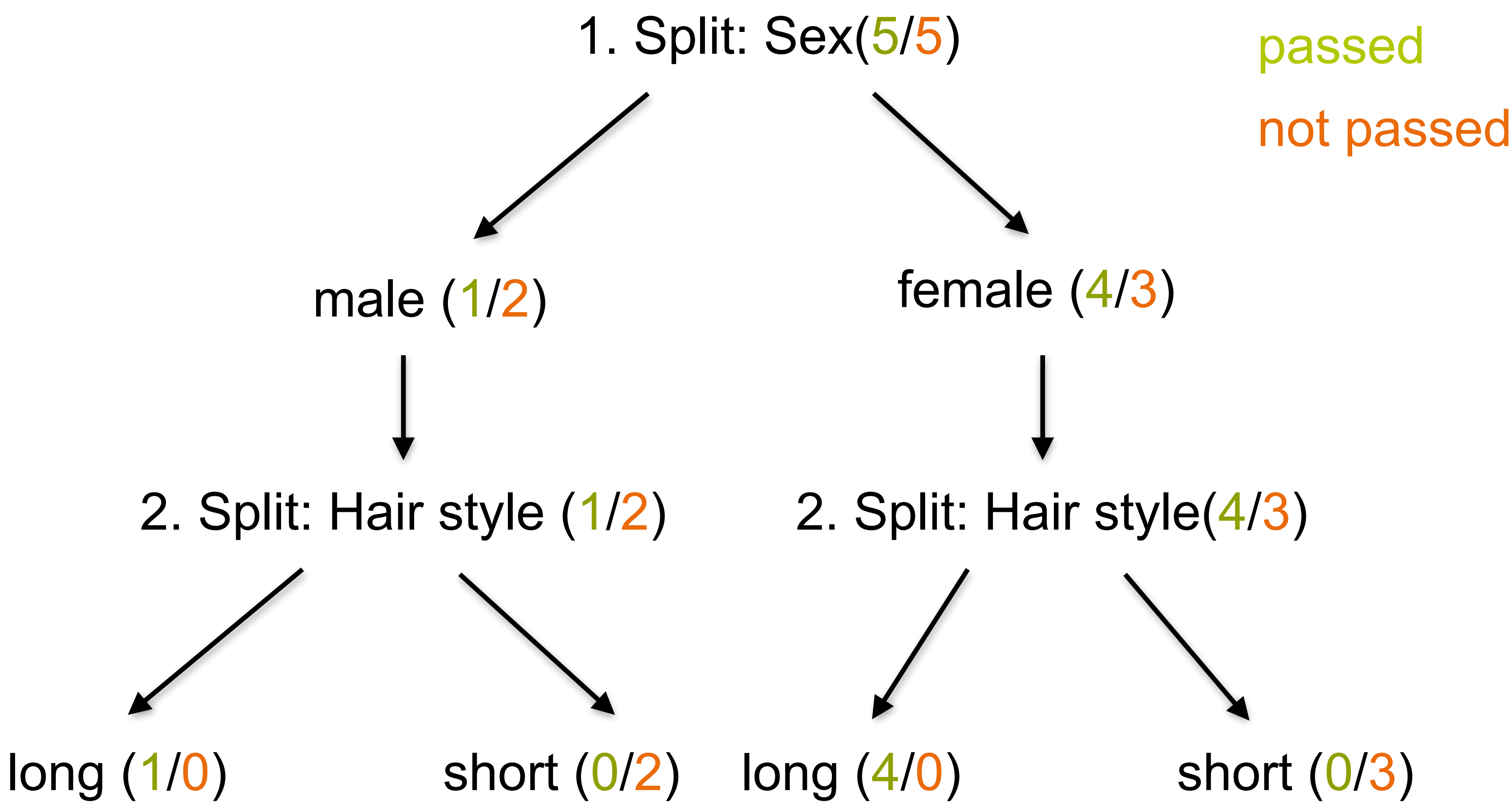
DECISION TREE: EXAMPLE

Student_ID	Sex	Hair style	exam (target)
1	male	short	not passed
2	male	long	passed
3	female	long	passed
4	male	short	not passed
5	female	long	passed
6	female	long	passed
7	female	long	passed
8	female	short	not passed
9	female	short	not passed
10	female	short	not passed

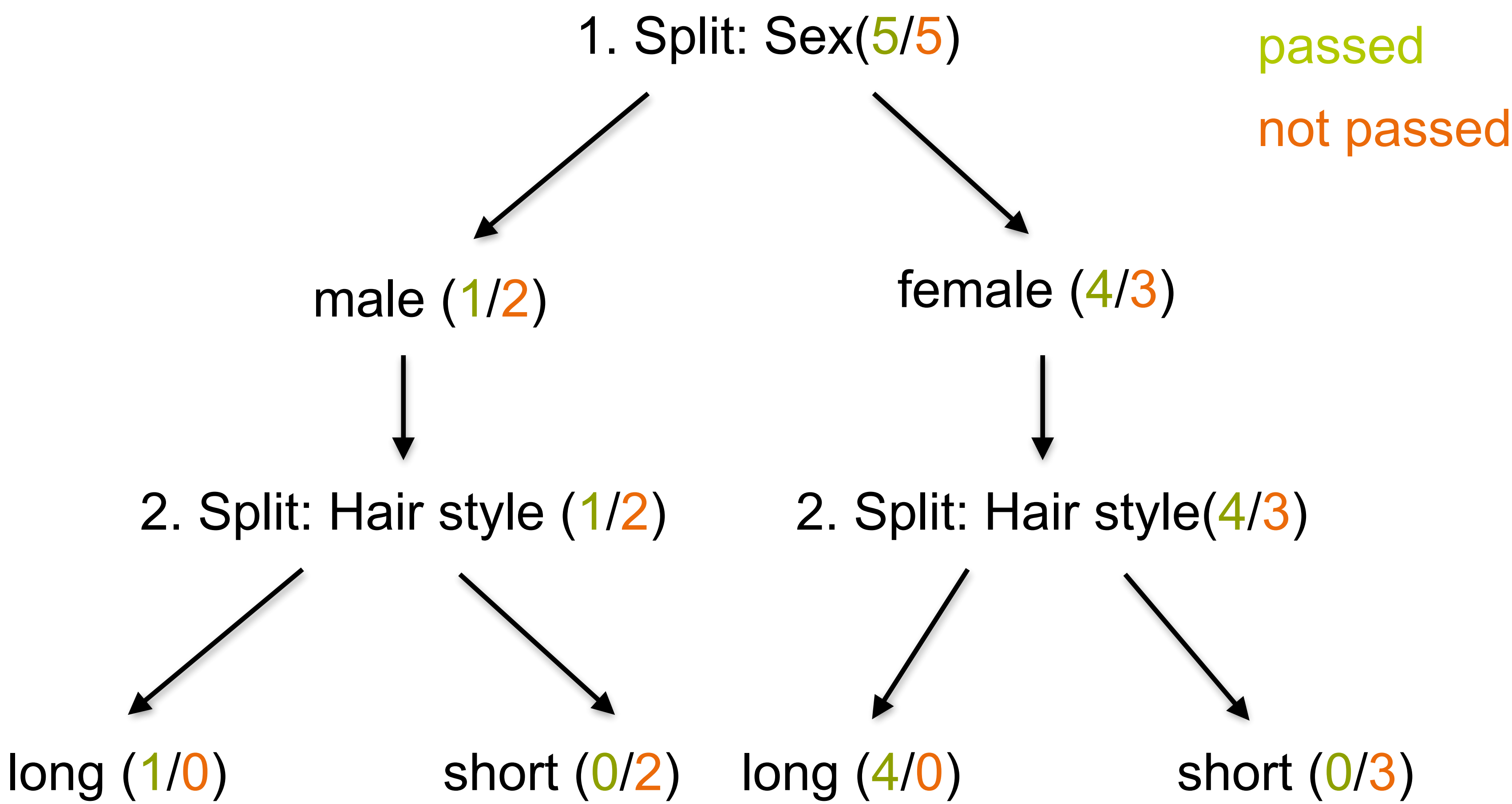
Features: Sex, Hair style

Target: exam (categorical)

DECISION TREE: EXAMPLE



DECISION TREES: EXAMPLE



Question: Can we do better?

DECISION TREES: EXAMPLE

1. Split: Hair style (5/5)

passed

not passed

long (5/0)

short (0/5)

Decision trees try to separate the data with as least splits as possible.

DECISION TREES: HOW TO SPLIT THE DATA?

- Gini impurity index (classification)
- Information Gain/Entropy (classification)
- Chi-Square (classification)
- Reduction in variance (regression)

DECISION TREES: GINI IMPURITY INDEX SPLIT (CLASSIFICATION)

$$G = \sum_{classes} p_i^2$$

1. Split: Sex(5/5)

passed

not passed

male (1/2)

$$G = \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 = 0.5$$

female (4/3)

$$G = \left(\frac{4}{7}\right)^2 + \left(\frac{3}{7}\right)^2 = 0.51$$

$$G = \frac{3}{10} * 0.5 + \frac{7}{10} * 0.51 = 0.507$$

DECISION TREES: GINI PURITY INDEX SPLIT (CLASSIFICATION)

$$G = \sum_{classes} p_i^2$$

1. Split: Hair Style (5/5)

passed

not passed

long (5/0)

$$G = 1^2 + 0^2 = 1$$

short (5/0)

$$G = 1^2 + 0^2 = 1$$

$$G = 1 * 0.5 + 1 * 0.5 = 1$$

DECISION TREES: GINI PURITY INDEX SPLIT (CLASSIFICATION)

$$G = \sum_{classes} p_i^2$$

1. Split: Hair Style (5/5)

passed

not passed

long (5/0)

$$G = 1^2 + 0^2 = 1$$

short (5/0)

$$G = 1^2 + 0^2 = 1$$

$$G = 1 * 0.5 + 1 * 0.5 = 1$$

1 > 0.507 => Split on hair style

DECISION TREES: VARIANCE REDUCTION (REGRESSION)

car_ID	PS	price (target)
1	300	30.000
2	400	32.000
3	425	36.000
4	450	48.000
5	600	60.000

1. Split: PS < 420

32.000
30.000
Var

60.000
48.000
36.000
Var

Weighted Variance

< oder >

2. Split: PS < 500

48.000
36.000
32.000
30.000
Var

60.000
Var

Weighted Variance

DECISION TREES: HYPERPARAMETERS

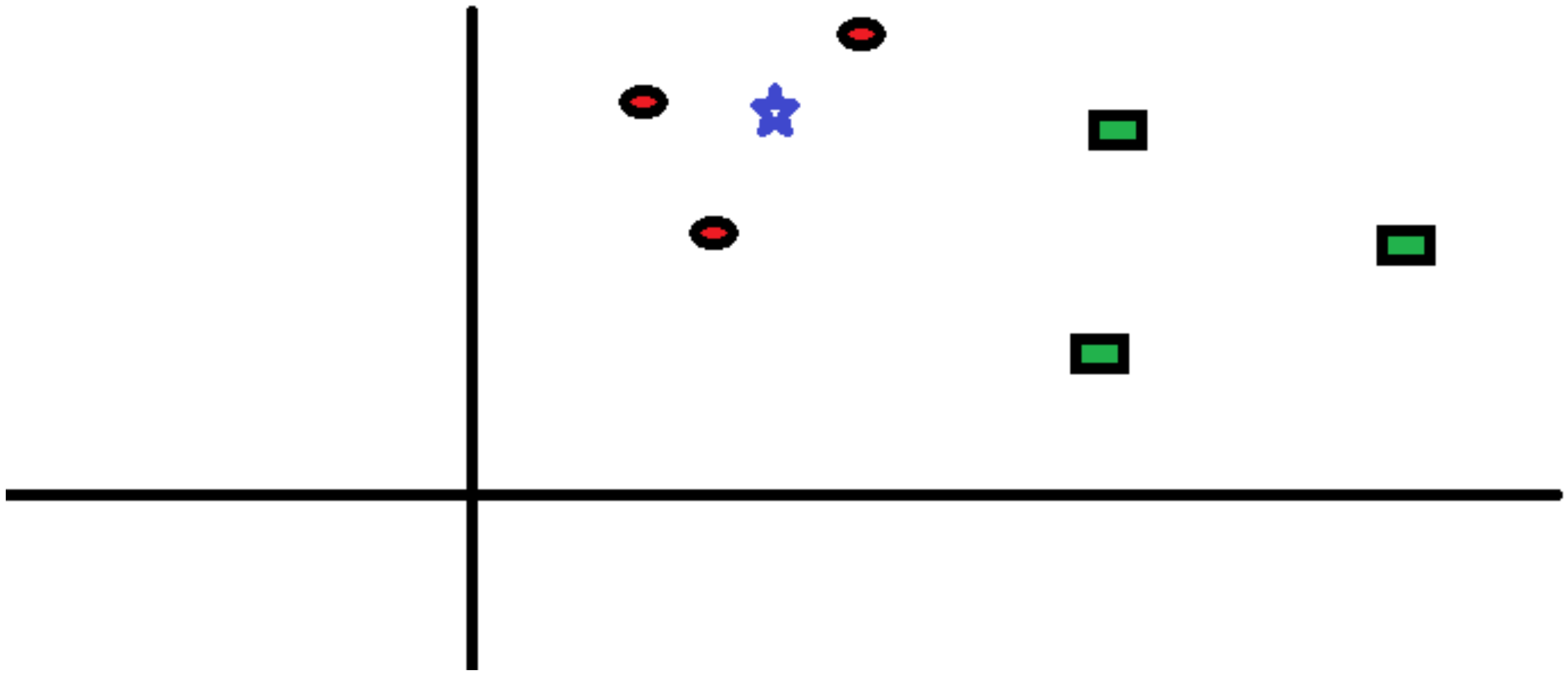
- Min_samples_split:
  Minimum samples per node before a split
- Min_sample_leaf:
  Minimum samples per leaf node after a split
- Max_depth:
  Maximum number of splits
- Max_features:
  Maximum number of splits to try for a each split

Demo: 04_Models (Trees)

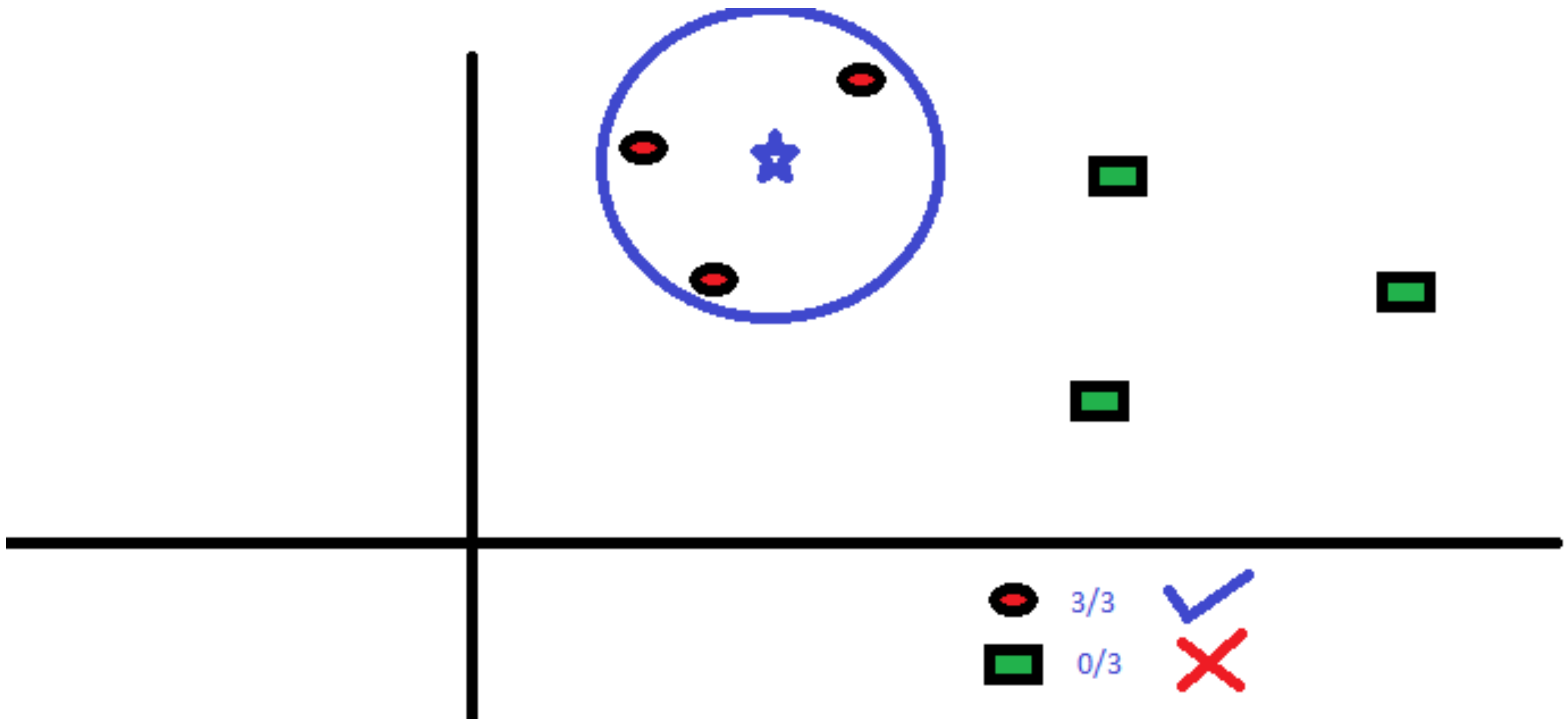
K NEAREST NEIGHBOR

- Classification and regression
- Easy to interpret
- Easy to understand
- Minimal training cost but expensive prediction
- Robust

K NEAREST NEIGHBOR: HOW DOES IT WORK?



K NEAREST NEIGHBOR: HOW DOES IT WORK?



K NEAREST NEIGHBOR: HYPERPARAMETERS

- Number of neighbors K
- Metric

Demo: 04_Models (KNN + Hyp. optimization)

ENSEMBLE METHODS

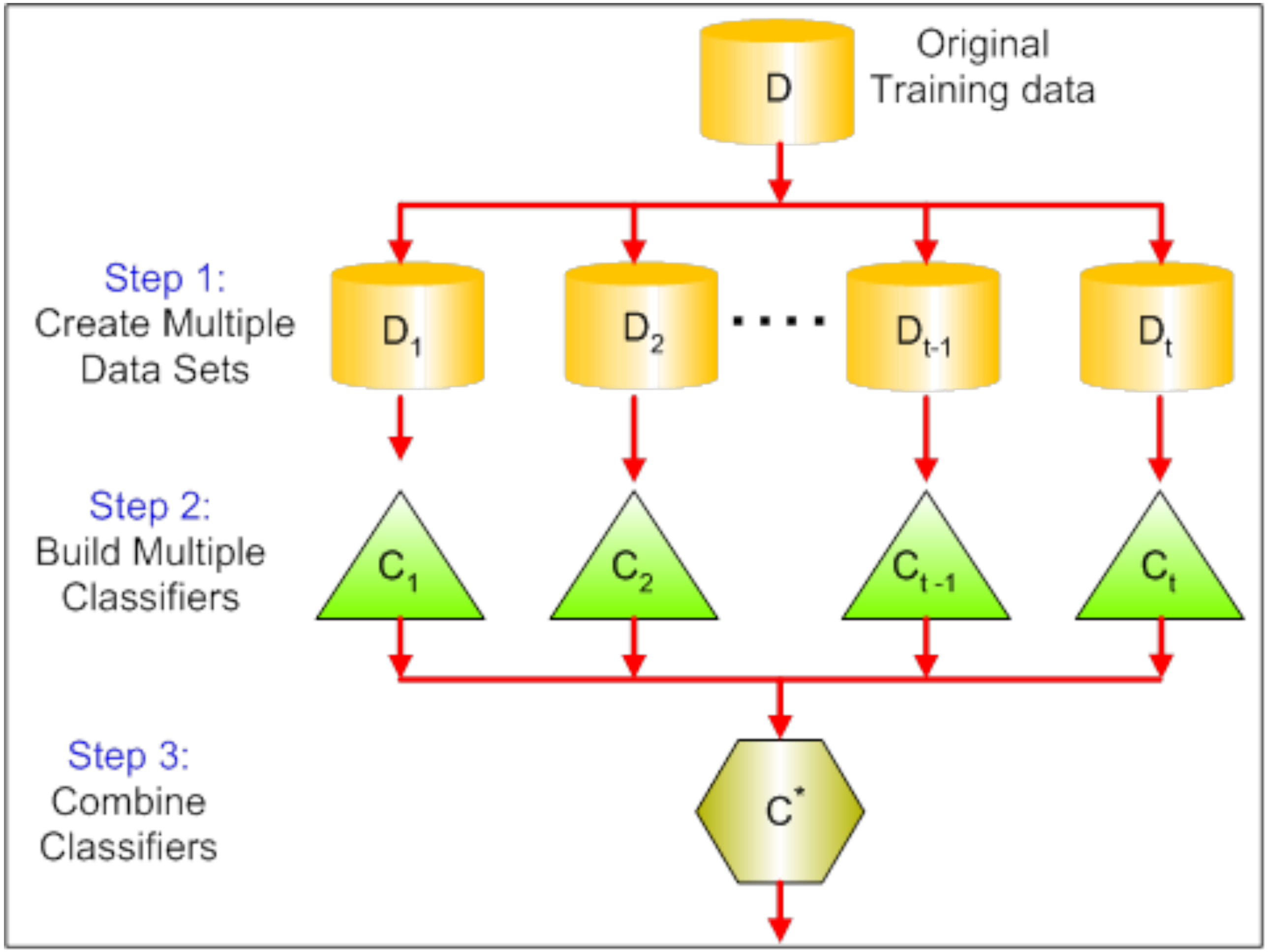
- Ensemble: Combination of several models
- Very powerful

- Prediction error: bias + variance + (noise)
- Bagging: variance reduction
- Boosting: bias reduction

BAGGING I

- Combines several independent models by averaging over their prediction results
- Reduces variance
- Works best with complex models (low bias)
- Example: RandomForest

BAGGING II

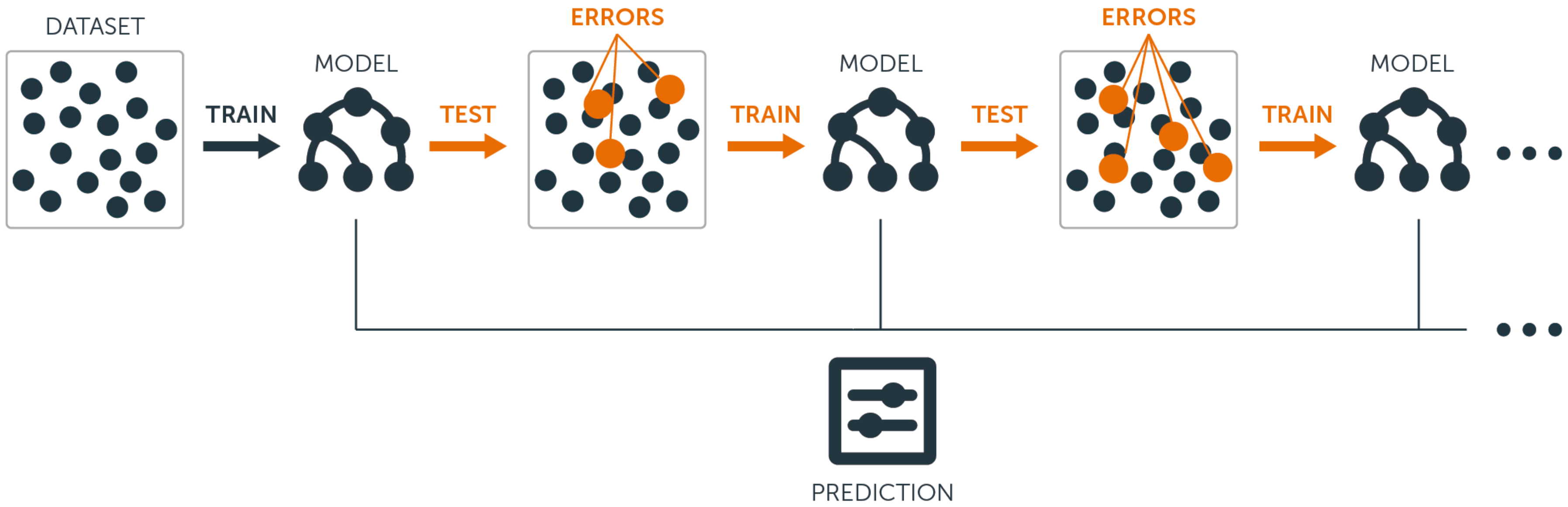


D
Original
Training data
Step 1:
Create Multiple
Data Sets
D1
D2
Dt-1
Dt
Step 2:
Build Multiple
Classifiers
C1
C2
Ct -1
Ct
Step 3:
Combine
Classifiers
C*

BOOSTING I

- Sequentially build models on top of each other while using the error of the previous model as the target of the new model
- Reduces bias
- Works best with weak models (low variance)
- Example: Gradient Boosting Decision Tree

BOOSTING II



Demo: 04_Models (Ensemble)

TEAM CHALLENGE: HOUSE PRICE PREDICTION

GrLivArea	GarageArea	…	SalePrice
100	35	…	128.000
150	45	…	254.000

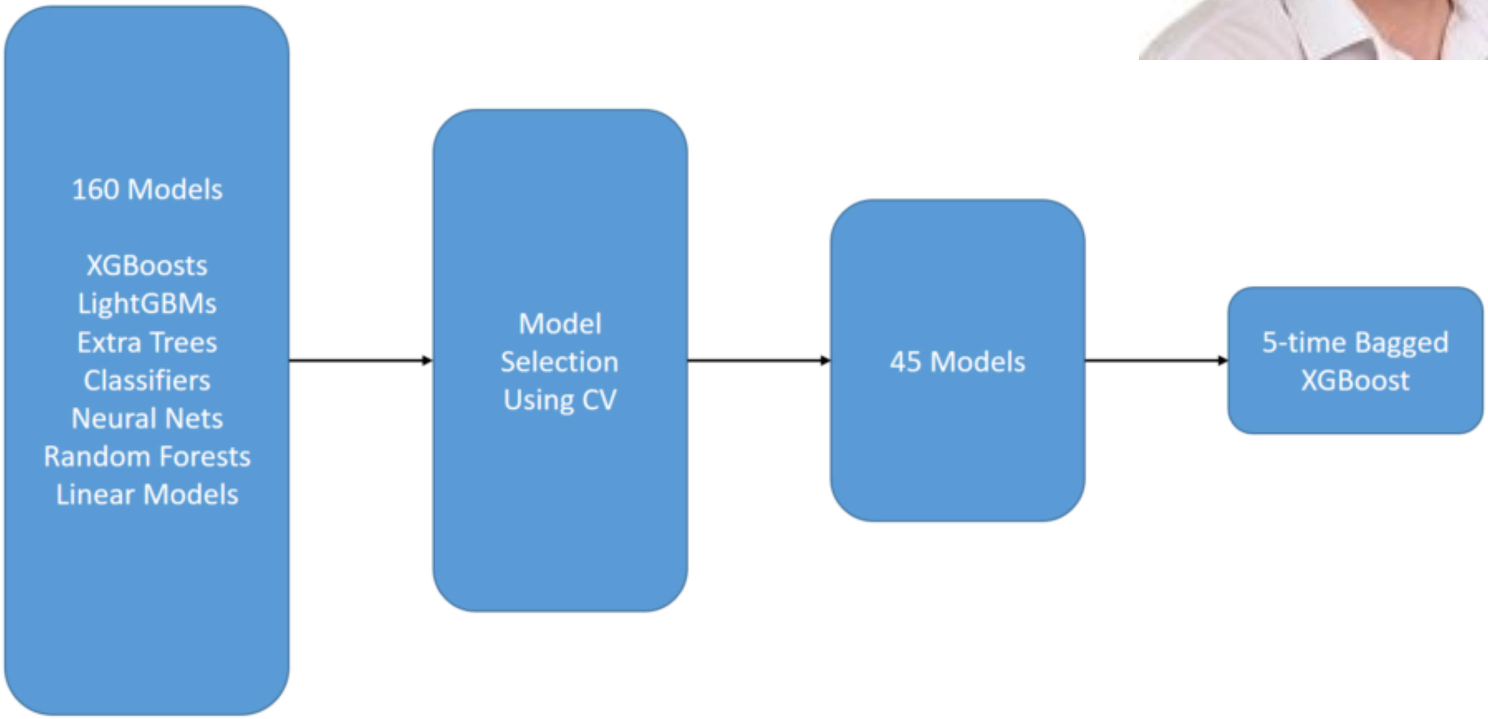
$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(log(prediction) - log(target))^2}$$

TEAM CHALLENGE: HOUSE PRICE PREDICTION

Name	Starting point	To do
Level 1	From scratch	Missing values EDA Feature Engineering Validation Model building
Level 2	~ 70 Features (semi cleaned)	Feature Engineering Validation Model building
Level 3	~200 Features (cleaned)	Model building Hyperparameter optimization

EXPERTS ON KAGGLE

- 80% Feature Engineering
- 20% Model building/tuning
- Ensemble methods



CAN WE USE KAGGLE AT OUR DEPARTMENT?

THANK YOU!