Universität Stuttgart
KI – Institute for Artificial Intelligence

Analytic Computing

Machine Learning
5 Classification and its Evaluation

Prof. Dr. Steffen Staab

Nadeen Fatallah
Daniel Frank
Akram Sadat Hosseini
Rodrigo Lopez
Osama Mohamed
Yi Wang
Tim Schneider

https://www.ki.uni-stuttgart.de/

Now all in different places!

- partially based on slides by
  - T. Gottron & M. Strohmaier, U. Koblenz-Landau
  - Florian Lemmerich et al, U. of Würzburg

http://west.uni-koblenz.de/en/studium/lehrveranstaltungen/ws1516/machine-learning-and-data-mining

Today‘s learning objectives (Monday, May 12, 2025)

Completing this slide deck you should know:

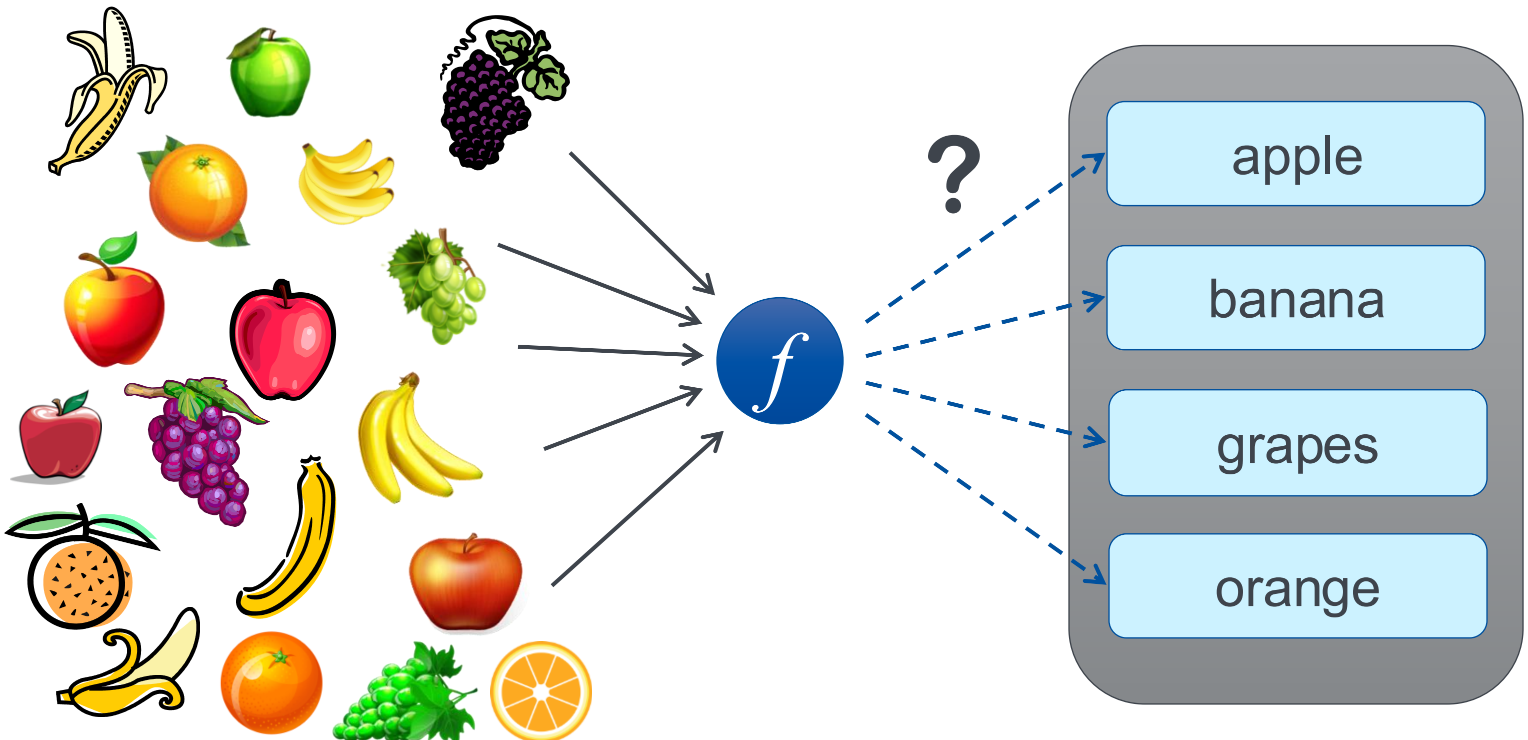
- What is *Classification*?
- What does it mean to *learn a classifier*?
- Machine Learning is optimization
  - In particular:
    A classifier is learned using optimization

Classification

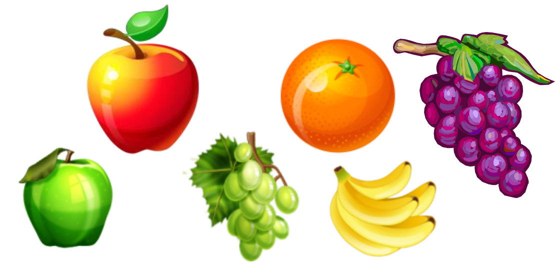
Example use cases for classification

Classification

Classification is a method that assigns labels to object representations



Ground Truth Classifier f



Each **object** $x_i \in X$ is described using a list of m attributes from a set $\mathcal{A} = \{A_1, \dots, A_m\}$

$$\forall i, k\colon\ x_i = \left(x_{i,1}, x_{i,2}, \dots, x_{i,m}\right)^T,$$

$$x_{i,k} \in A_k$$

(green, round, smooth)

(orange, round, rough)

Category labels

$$\mathrm{Y} = \{l_1, \dots, l_k\}$$

{apple, banana, grapes, orange}

The classifier f is a function

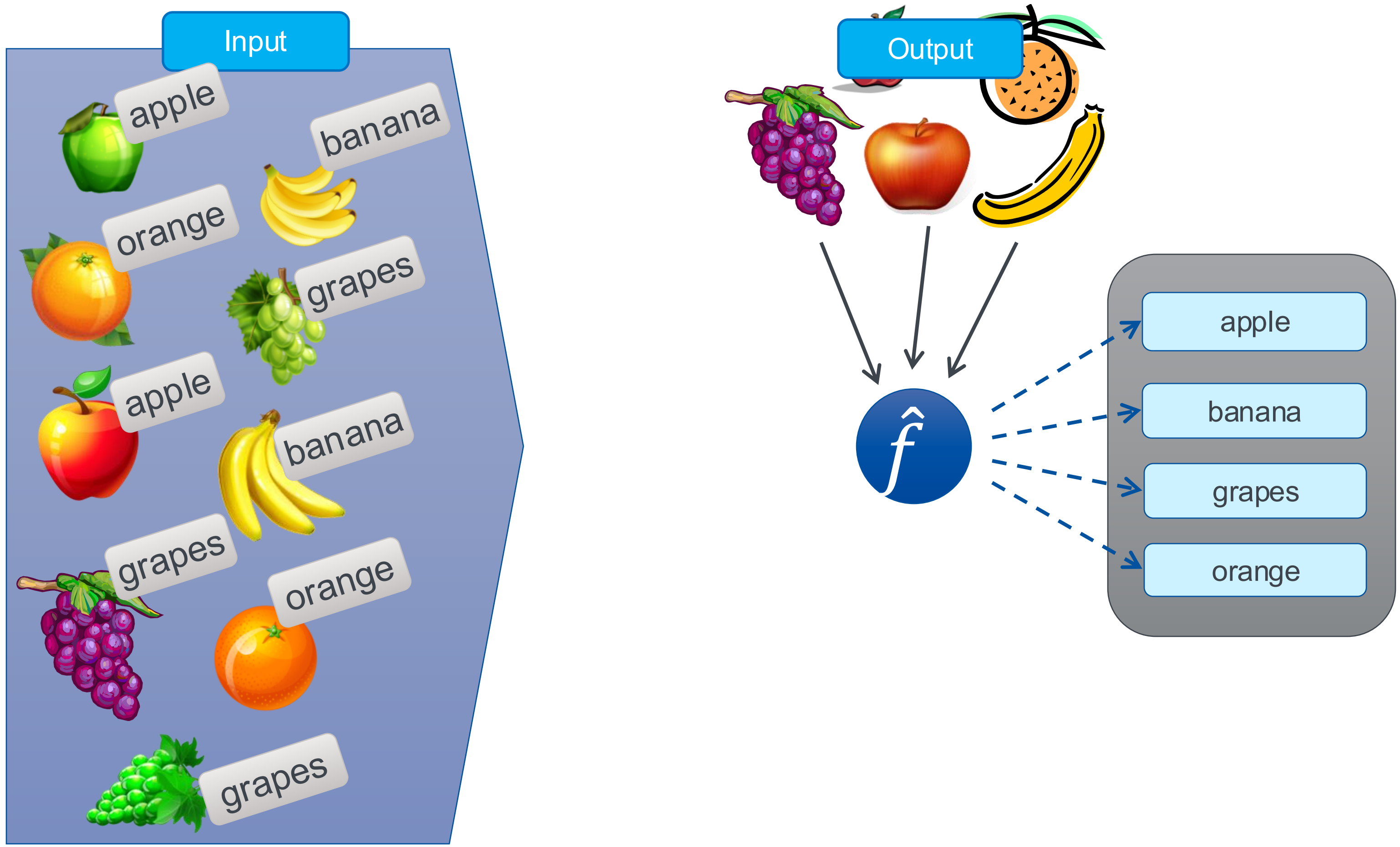
that maps descriptions of objects

onto category labels

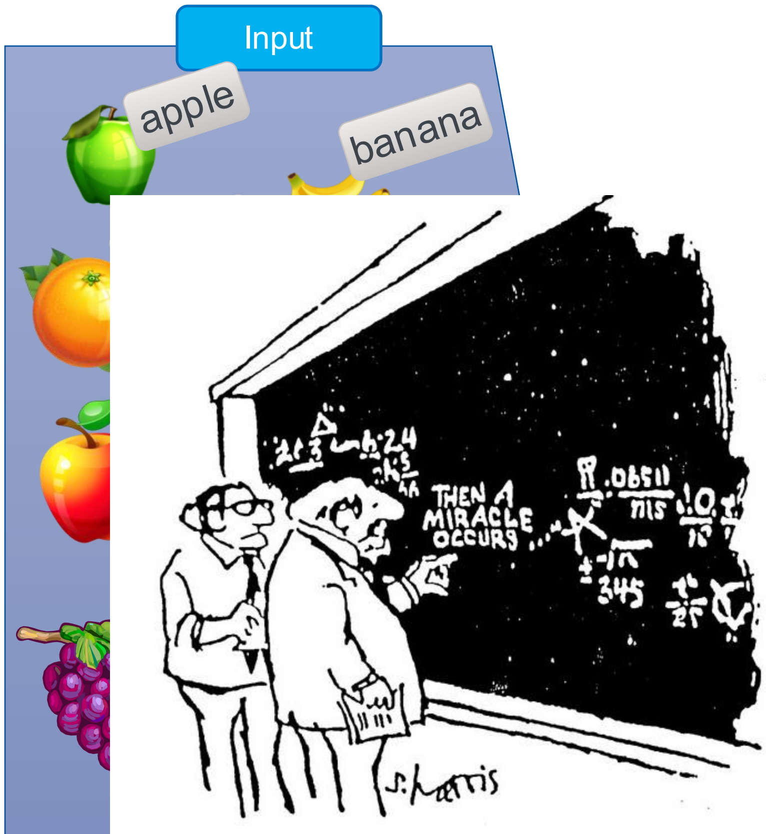
$$f\colon X \to Y$$

Many ways to approximate f

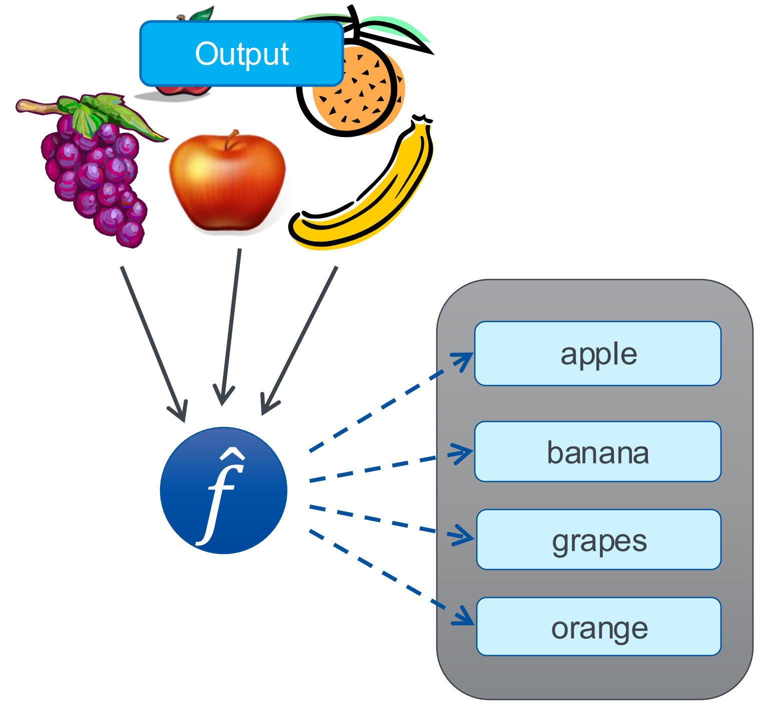
Learning a classifier with pre-classified training data (labeled training data)



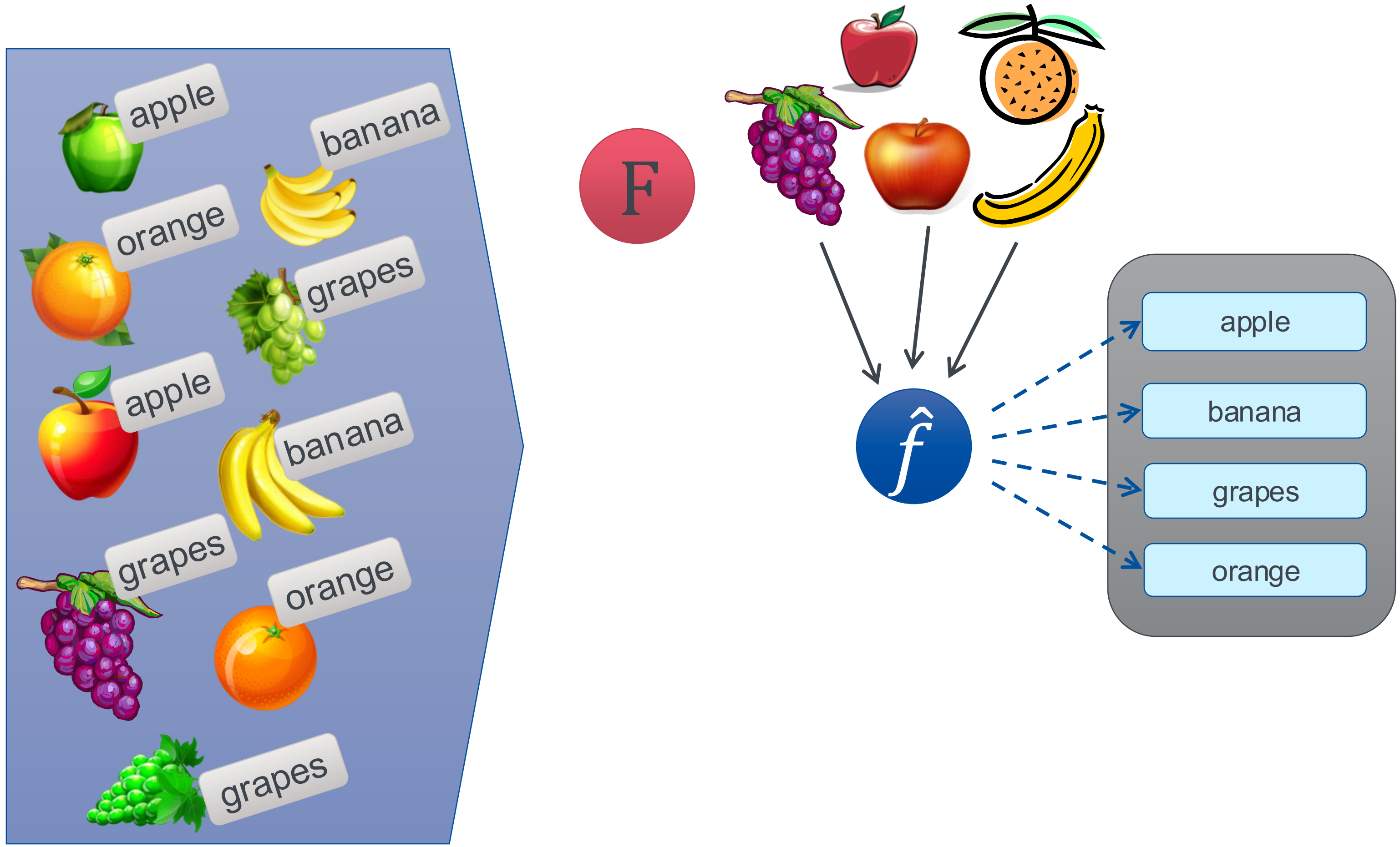
Learning a classifier with pre-classified training data (labeled training data)



"I think you should be more explicit here in Step Two."
By Sidney Harris, Copyright 2007, The New Yorker



Learning a classifier with pre-classified training data (labeled training data)



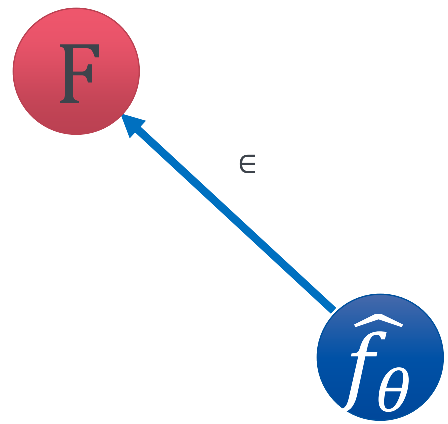
Families of functions $\Gamma = \{f | \ldots\}$

Which families of functions do you know?

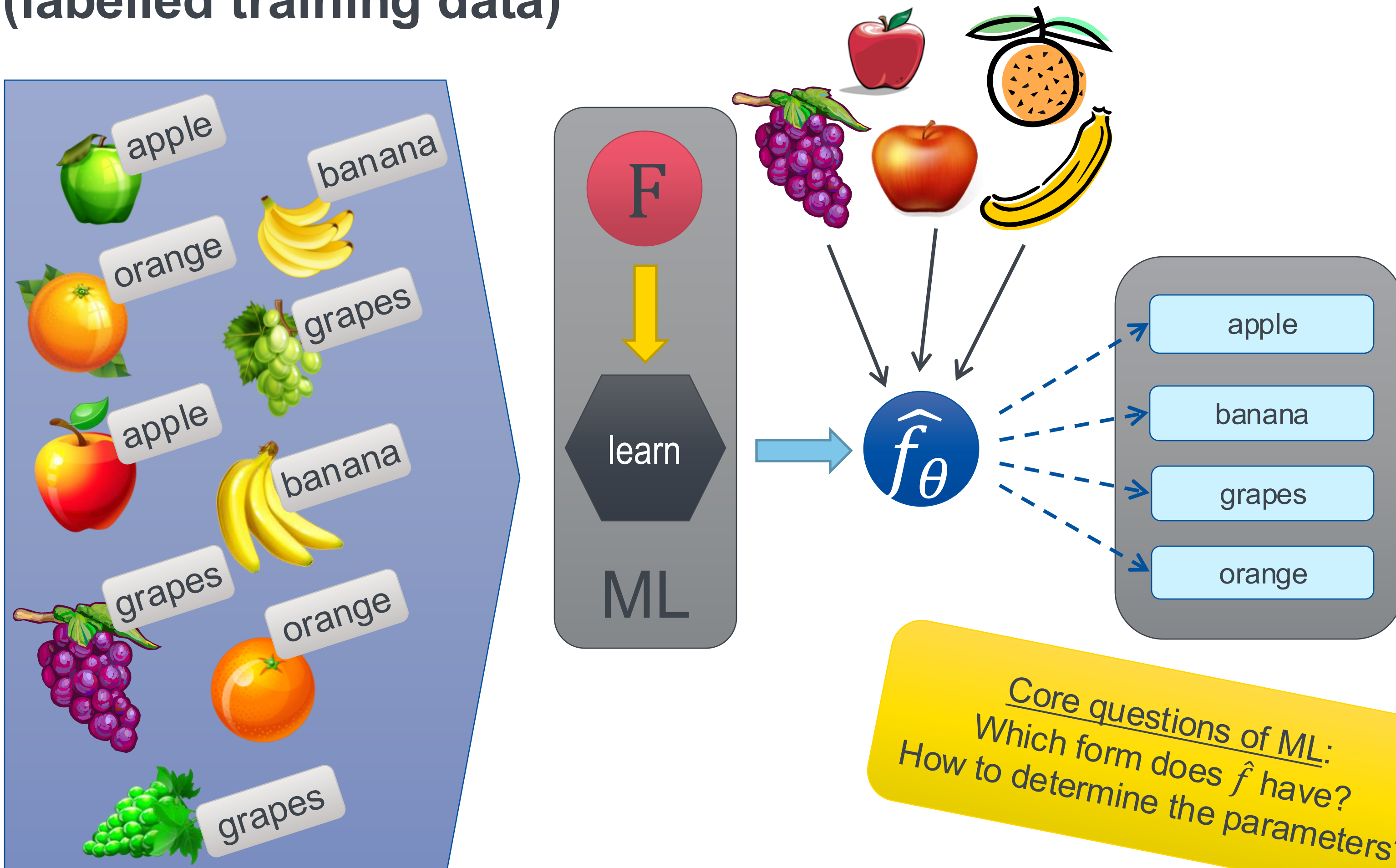
- family of constant functions
  $$\{f | f(x) = c\}$$
- family of linear functions
  $$\{f | f(x) = ax + c\}$$
- exponential functions
  $$\{f | f(x) = a^x\}$$
- ... $\hat{f}$

what characterizes a family of functions?

- form
- parameters $(a, b, c, \ldots \theta_1, \ldots \theta_t)$



Learning a classifier with pre-classified training data (labelled training data)



<u>Core questions of ML</u>:
Which form does $\hat{f}$ have?
How to determine the parameters?

Defining the task of learning a classifier

Given dataset $\mathcal{D}$ with n objects $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \sim X \times Y$

each described using m attributes

and an observed category label y_i

$$\forall i, k: \; x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m}),$$

$$x_{i,k} \in A_k,$$

$$y_i \in L$$

where $X \times Y$ is the joint distribution of random variables X and Y

apple

(green, round, smooth)

(orange, round, rough)

orange

Given a family of functions F,

find a function $\widehat{f_\theta} \in \mathrm{F}$,

such that $\widehat{f_\theta}: A_1 \times \cdots \times A_m \to Y$

and $\widehat{f_\theta}$ minimizes the *empirical risk* ("loss")

on observed data (X,Y)

$$\widehat{f_\theta} = \underset{\tilde{f} \in \mathrm{F}}{\operatorname{argmin}} \sum_{i=1\dots n} \ell(\tilde{f}(x_i), y_i)$$

$\widehat{f_\theta}(x_i)$ should reproduce observation y_i

Empirical risk minimization and overfitting

Minimizing loss:

$$\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{x,y\sim P(X,Y)}[\ell(\widehat{f_\theta}(x), y)]$$

Theory of machine learning:

Law of large numbers – minimzing empirical risk:

$$\underset{\theta}{\operatorname{argmin}}\, \frac{1}{N}\sum_{i=1}^{N} \ell(\widehat{f_\theta}(x_n), y_n) \quad \xrightarrow{\text{“}LLN\text{“}} \quad \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}[\ell(\widehat{f_\theta}(x), y)]$$

Viewing machine learning as empirical risk minimization

Designing models:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{f}_\theta(x_i), y_i)$$

Choosing/augmenting data:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{f}_\theta(x_i), y_i)$$

Designing loss function:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{f}_\theta(x_i), y_i)$$

Designing optimization methods:

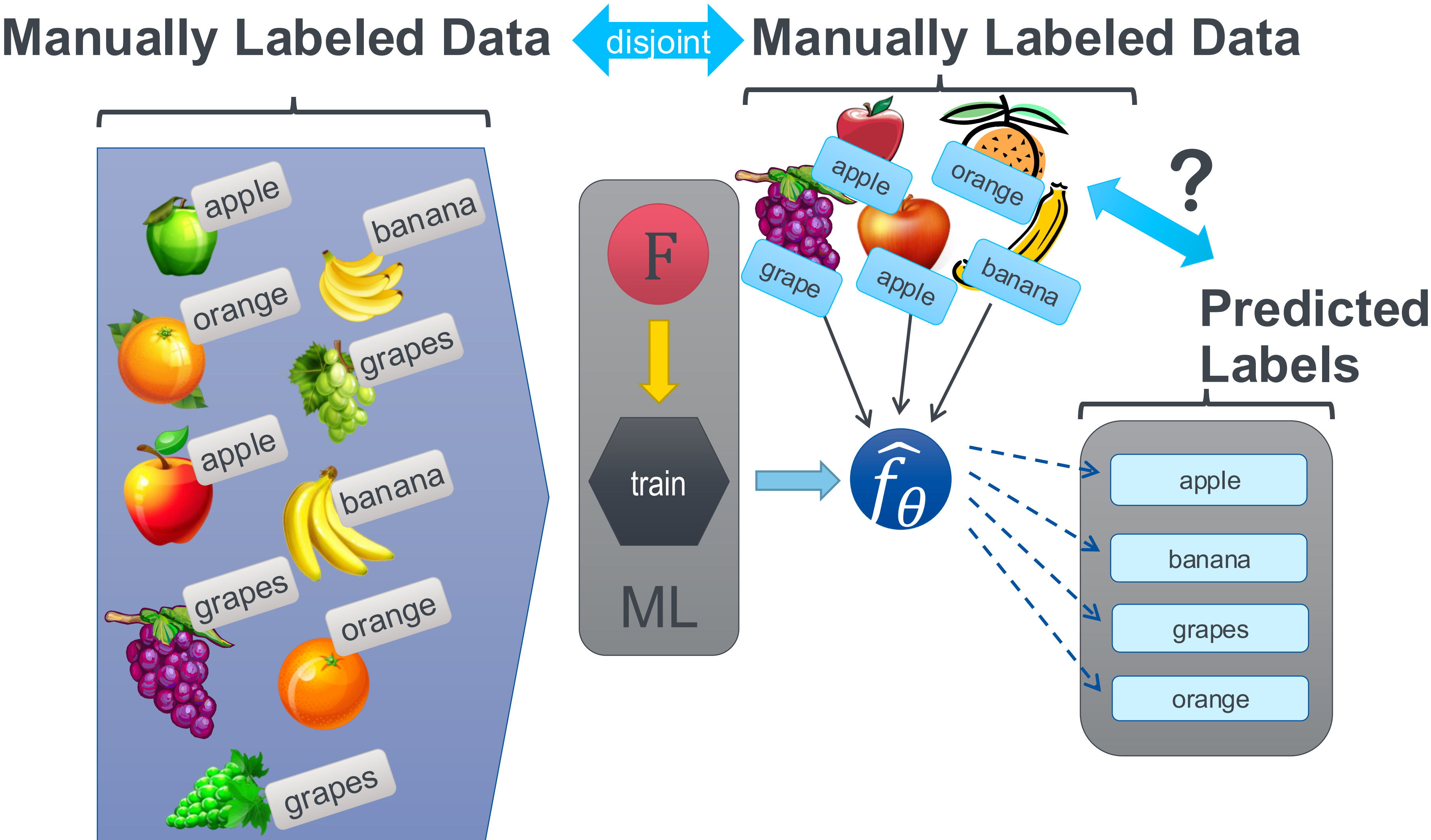
$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{f}_\theta(x_i), y_i)$$

A Few Useful Things to Know about Machine Learning

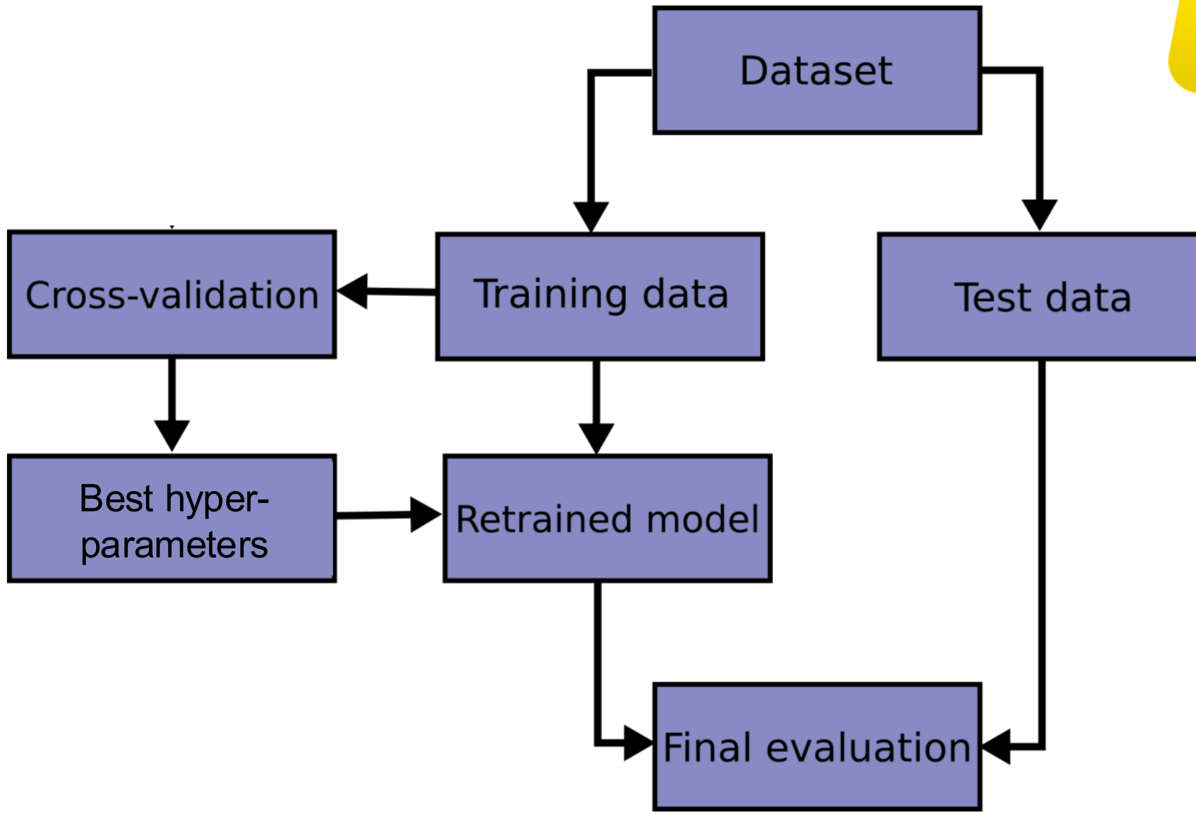
- learning = data + representation + evaluation + optimization
  - data: quality of data
  - representation: which form does $\hat{f}_\theta$ have?
  - optimization: how to determine parameters θ?
  - evaluation: empirical risk/loss

Validation and Evaluation

Evaluation using Labeled Data



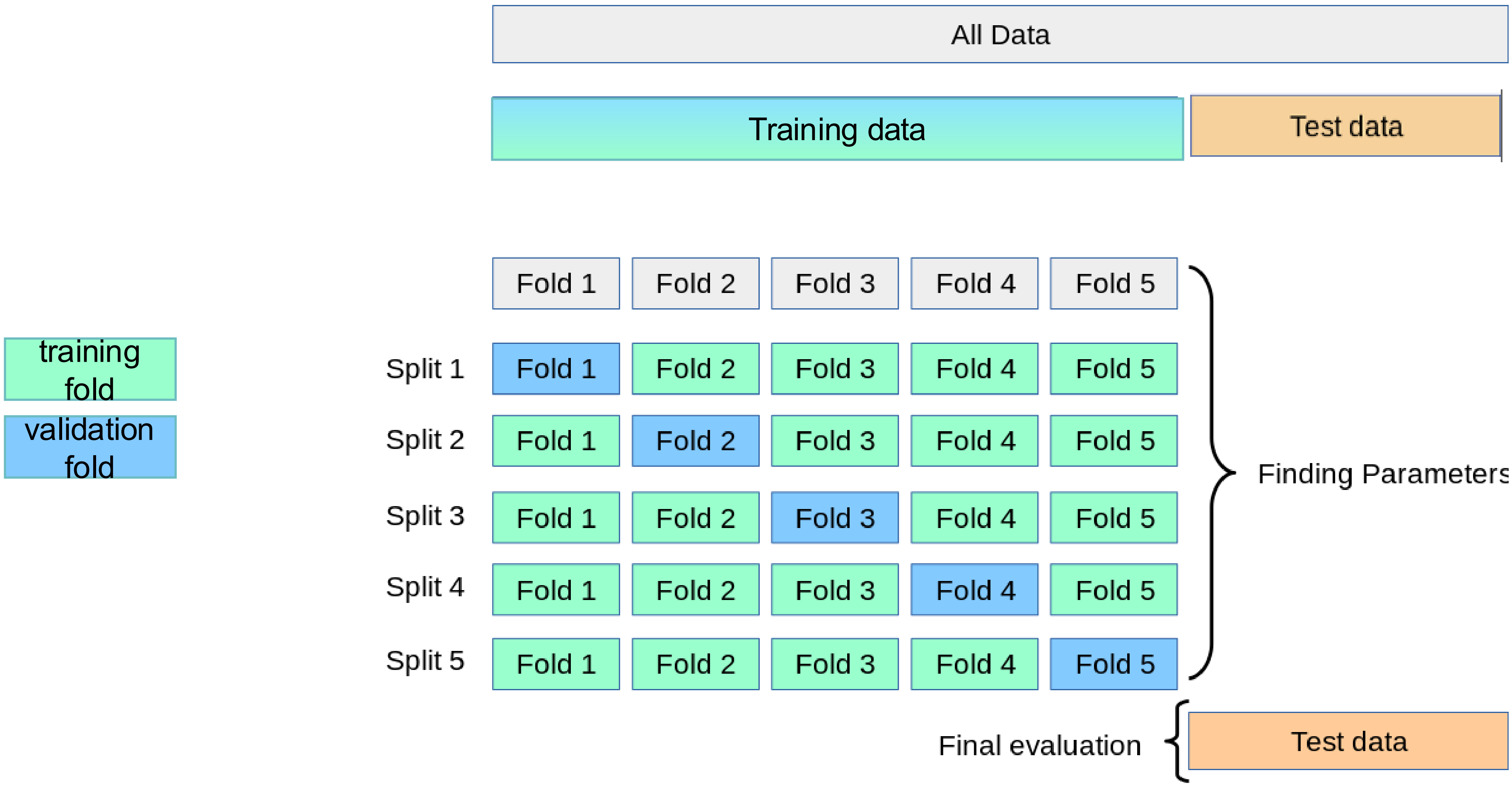
Keep test data separate from your training efforts



Otherwise no chance to detect overfitting

Modified version from https://scikit-learn.org/stable/modules/cross_validation.html

Split Non-test Data into Training Data and Validation Data and k-fold cross-validation



Modified version from https://scikit-learn.org/stable/modules/cross_validation.html

Fighting overfitting when evaluating

- At each step: three non-overlapping sets
  - $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 = \mathcal{D},\ \mathcal{D}_1 \cap \mathcal{D}_2 = \mathcal{D}_1 \cap \mathcal{D}_3 = \mathcal{D}_2 \cap \mathcal{D}_3 = \emptyset$
- Train to minimize $\sum_{(x,y)\in\mathcal{D}_1} \ell\left(\hat{f}(x), y\right)$
- Validate results with $\sum_{(x,y)\in\mathcal{D}_2} \ell\left(\hat{f}(x), y\right)$
  - determine best hyperparameters, e.g. k in kNN
- Evaluate results with $\sum_{(x,y)\in\mathcal{D}_3} \ell\left(\hat{f}(x), y\right)$
  - determine how well you do
    after modeling and optimization

Training data, validation data and evaluation data **must not overlap** in pairwise intersection

How to evaluate?

- At each step: three non-overlapping sets
  - $\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 = \mathcal{D},\ \mathcal{D}_1 \cap \mathcal{D}_2 = \mathcal{D}_1 \cap \mathcal{D}_3 = \mathcal{D}_2 \cap \mathcal{D}_3 = \emptyset$
- Train to minimize $\sum_{(x,y)\in\mathcal{D}_1} \ell\left(\hat{f}(x), y\right)$
- Validate results with $\sum_{(x,y)\in\mathcal{D}_2} \ell\left(\hat{f}(x), y\right)$
  - determine best hyperparameters, e.g. k in kNN
- Evaluate results with $\sum_{(x,y)\in\mathcal{D}_3} \ell\left(\hat{f}(x), y\right)$
  - determine how well you do after modeling and optimization

task: guide parameter learning

Same loss function? **No**

task: inform engineer or user about quality of system

User-oriented Evaluation of Classifiers

Confusion Matrix Representing Quality of System

- Extend confusion matrix to multiple categories

		Ground truth				
		c_1	c_2	c_3	…	c_J
$\hat{f}_\theta$	c_1	*correct*	error	error		error
	c_2	error	*correct*	error		error
	c_3	error	error	*correct*		error
	…				…	
	c_J	error	error	error		*correct*

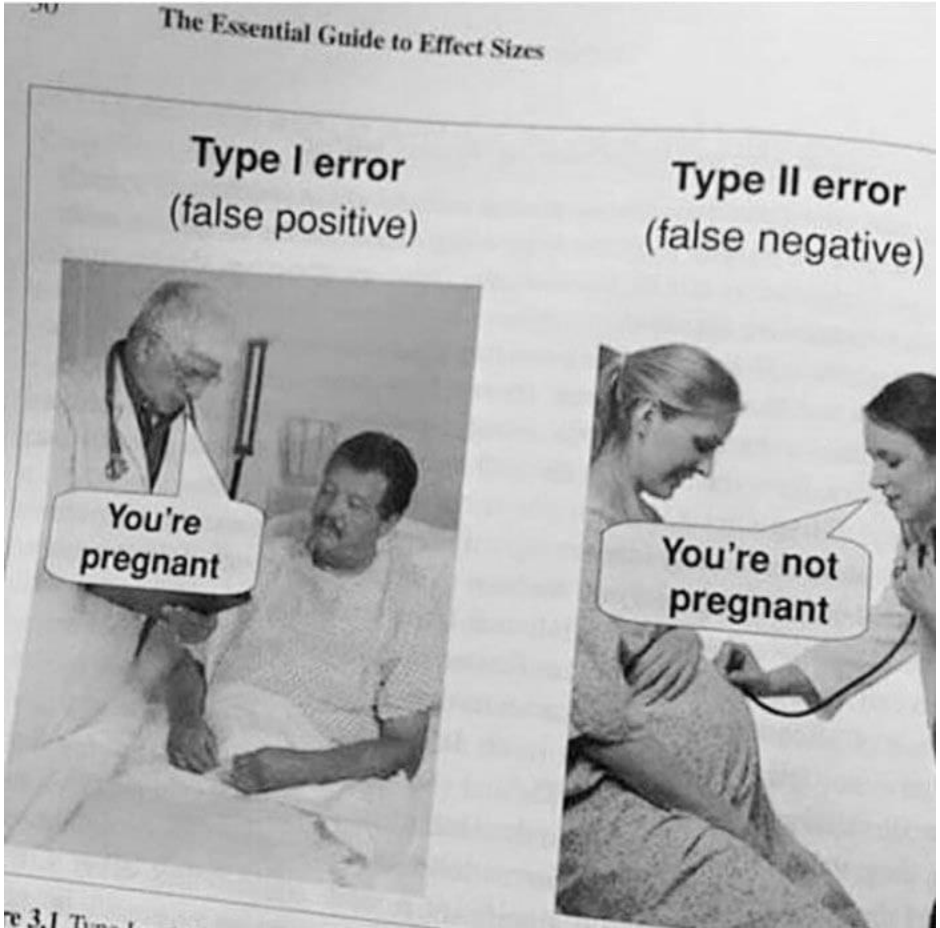
- Metrics for each category:
  - Recall: $$r(c_i) = \frac{a_{ii}}{\sum_{k=1}^{J} a_{ki}}$$
  - Precision: $$p(c_i) = \frac{a_{ii}}{\sum_{k=1}^{J} a_{ik}}$$
  - F_1: harmonic mean of recall and precision: $$F_1 = \frac{2 \cdot r \cdot p}{r + p}$$

Ratio of how many objects in c_i have been correctly classified

Ratio of how many objects have been correctly classified as c_i

A bit of both ...

Type I vs Type II error



Confusion Matrix

- Extend confusion matrix to multiple categories

		Ground truth				
		c_1	c_2	c_3	…	c_J
$\hat{f}_\theta$	c_1	*correct*	error	error		error
	c_2	error	*correct*	error		error
	c_3	error	error	*correct*		error
	…				…	
	c_J	error	error	error		*correct*

- Metrics:
  - Globally: Accuracy, 0-1-loss

$$acc = \frac{\sum_{i=1}^{J} a_{ii}}{\sum_{i=1}^{J} \sum_{j=1}^{J} a_{ij}}$$

Ratio of correct decisions

Example

Evaluating a Mushroom Classifier

- Three categories: poisonous, edible, psychoactive
  - Confusion matrix

		Ground truth			
		Poisonous	Edible	Psycho-active	***Total***
$\hat{f}_\theta$	Poisonous	*5*	1	2	***8***
	Edible	2	*10*	4	***16***
	Psycho-active	0	0	*6*	***6***
	Total	***7***	***11***	***12***	***30***

- For „Poisonous“: Recall, Precision, F_1?

$$r = \frac{5}{7} = 0.71 \quad p = \frac{5}{8} = 0.63 \quad F_1 = \frac{2 \cdot r \cdot p}{r + p} = 0.67$$

- Accuracy, 0-1-loss? $$acc = \frac{21}{30} = 0.7$$

micro/macro/weighted

Data Leakage

Data leakage

Example: Classify social media posts into hate speech or not

Tweet 1	Miller
Tweet 2	Smith
Tweet 3	Miller
Tweet 4	Smith
Tweet 5	McGuinness

All Data

Training data	Test data
Tweet 1	Tweet 3
Tweet 2	Tweet 4
Tweet 5	

In this setup, one does not learn what hate speech is, but how Smith and Miller tweet


THE WEB CONFERENCE
ACM
Using diversity as a source of scientific innovation for the Web
Barbara Poblete
Department of Computer Science, University of Chile
Amazon Visiting Academic*
*content not associated to work done at Ama

Experimental validation problems

Data leakage in model training/testing:
- Use of complete dataset in training phase (word embedding generation)
- Oversampling of HS class before train/test split

Data bias, surfaced by user-level analysis:
- 3 users responsible for 90% of HS
- 1 user generated 80% of HS data

"Hate speech detection is not as easy as you may think: A closer look at model validation" by Arango, Perez & Poblete (SIGIR 2019) *Extended Version, 2022.*

By fixing experimental issues, performance dropped to ~51% F1 (from ~93%)

Very close to our original Spanish baseline

Focus on the data is just as important as focus on performance metrics

As models become more obscure w/DL, experimental validation is key

Hate speech detection is not as easy as you may think: A closer ook at model validation" by Arango, Perez & Poblete (2019)
xtended Version, 2022.

Summary on Evaluation

- Evaluation is motivated by application
  - For example: you care about correct classification,
    you do not care whether you could have been almost correct
- A loss function is usually not a good evaluation function
  - As we will see later: A loss function needs to support the determination of parameters θ

Miscellaneous

Variations of the Classification Task

- Category types:
  - Flat vs. hierarchical
  - Exclusive vs. multiple categories
- Function $\widehat{f}_\theta$
  - Hard vs. soft assignments
  - Manual provision vs. machine learning
- Purpose
  - Descriptive Modelling
    - Explain the data
  - Predictive Modelling
    - Classify new data

Sources

- Optical recognition of fruits: http://de.ids-imaging.com/case-studies.html
- Document classification: http://www.ndm.net/opentext/capture-and-recognition/capture-center
- Email spam: http://www.gfi.com/blog/spam-emails-bringing-excitement-1978/
- Amanita muscaria: http://commons.wikimedia.org/wiki/File:Amanita_muscaria_3_vliegenzwammen_op_rij.jpg, CC-BY-SA 3.0-nl, Onderwijsgek
- Lactarius indigo: http://commons.wikimedia.org/wiki/File:Lactarius_indigo_48568.jpg, CC-BY-SA 3.0, Dan Molter

Universität Stuttgart
IPVS

Thank you!

Steffen Staab

E-Mail Steffen.staab@ipvs.uni-stuttgart.de

Telefon +49 (0) 711 685-To be defined

www.ipvs.uni-stuttgart.de/departments/ac/

Universität Stuttgart
Analytic Computing, IPVS
Universitätsstraße 32, 50569 Stuttgart