Lecture

Foundations of Artificial Intelligence

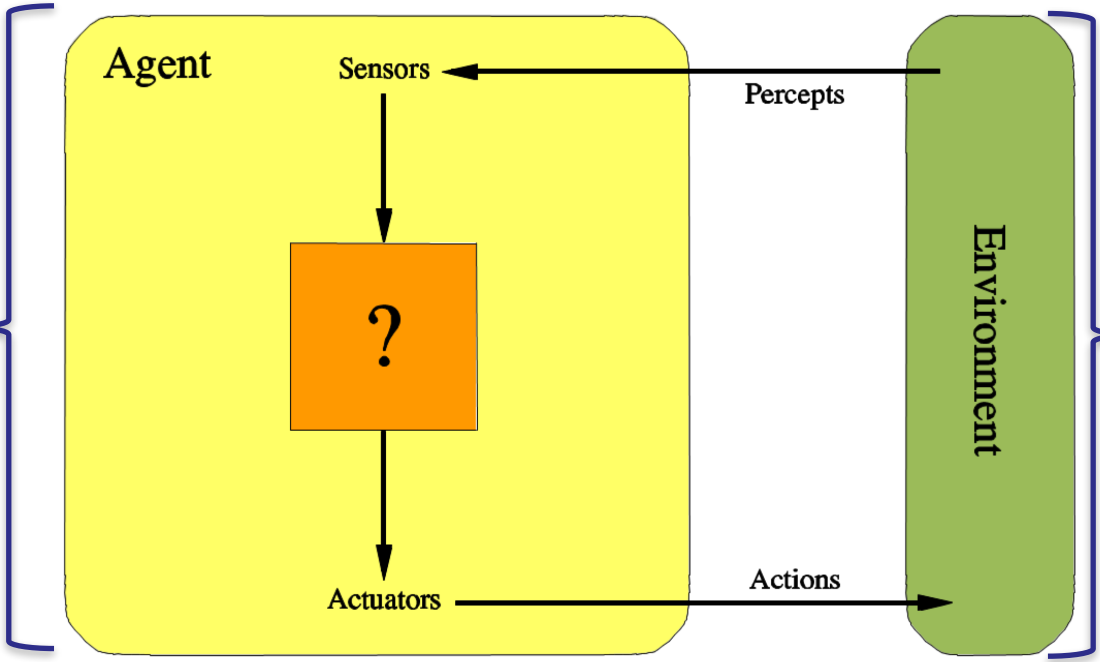
Part 8 – Machine Learning

Dr. Mohsen Mesgar

Universität Duisburg-Essen

Recall

Solution



Problem

Recall …

- For optimal path finding and optimal solution finding problems, we defined several algorithms to find the optimal path or solution
- These problems are mostly deterministic and fully observable
- However, we know that many environments are stochastic and partially observable
- These environment include a lot of uncertainty
- Agents need to deal with such uncertainty

Recall ...

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- One way to deal with uncertainty is to use probabilities
- One way to estimate probability of an event is to count the frequency of that event
- Prior probability: P(A)
- Posterior (Conditional) probability: $P(A \mid B)$
  - The probability of A when we have the knowledge that B already happened.
- Joint probability: P(A,B) = P(A)P(A|B)

- Bayes' rule:

$$\mathbf{P}(\mathbf{A} \mid \mathbf{B}) = \frac{\mathbf{P}(\mathbf{B} \mid \mathbf{A}) * \mathbf{P}(\mathbf{A})}{\mathbf{P}(\mathbf{B})}$$

Any other open questions?

Today

Machine Learning (ML)

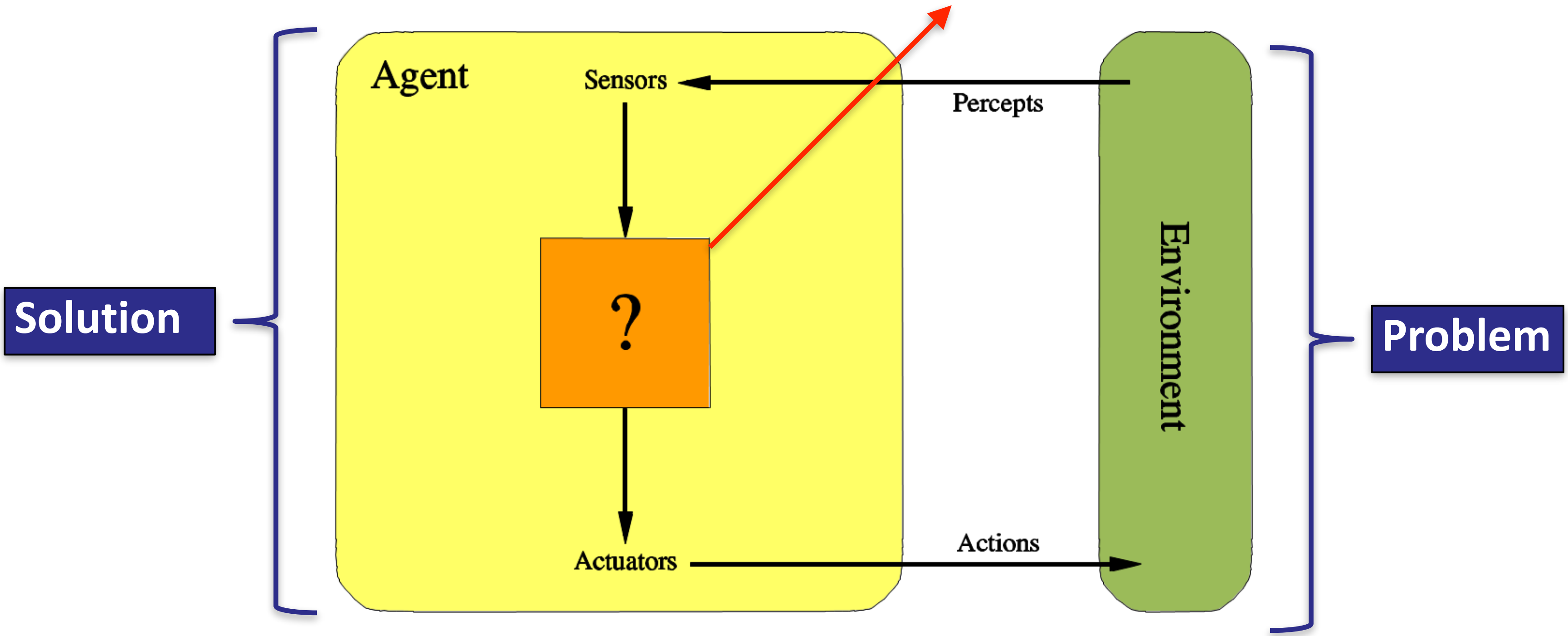
ML

- An agent should ideally behave like humans. We as humans can
  - deal with uncertainty,
  - learn to conduct a task using some supervision
- Since we can do it, AI agents should be able so
- However, the agents we discussed in previous lectures
  - fail when uncertainty is involved because they assume the environment is fully observable and deterministic
  - need humans to design algorithms that find solutions for problems
- In ML, we aim for agents that learn to gather the knowledge required for solving a problem, especially when uncertainty exists

ML

- ML lets agents **learn how to deal with uncertain environments.**
- In fact, agents **learn to optimize their knowledge** to achieve the predefined goals.

ML algorithms find this function that solves the problem

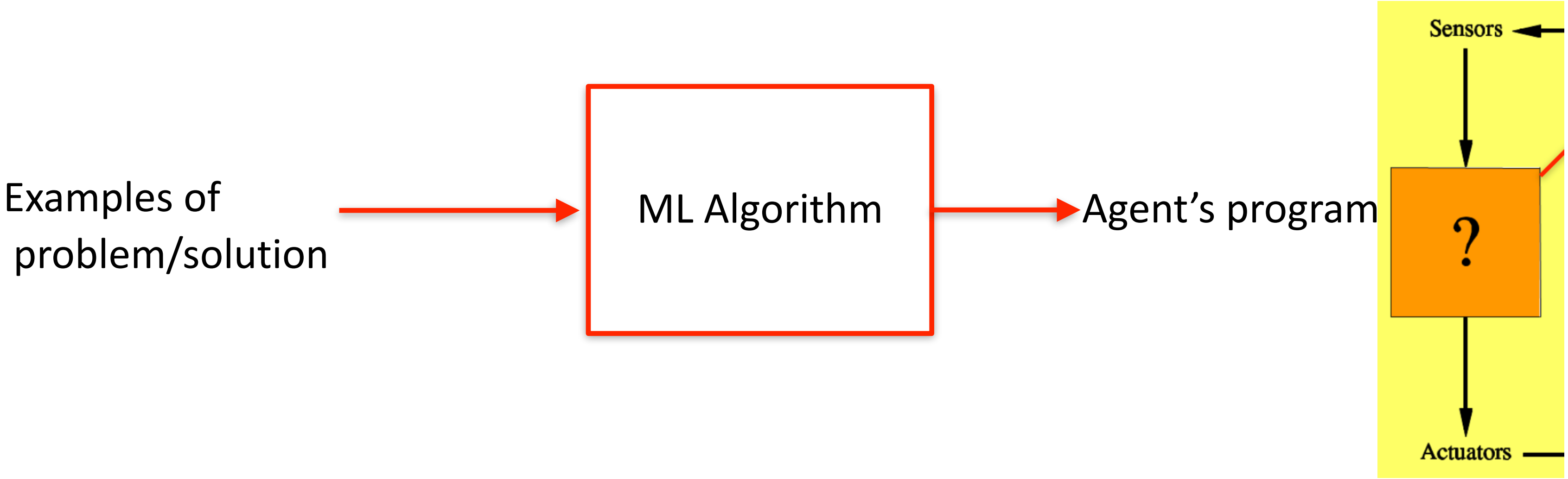


ML

- ML lets agents **learn how to deal with uncertain environments.**
- In fact, agents **learn to optimize their knowledge** to achieve the predefined goals.

ML

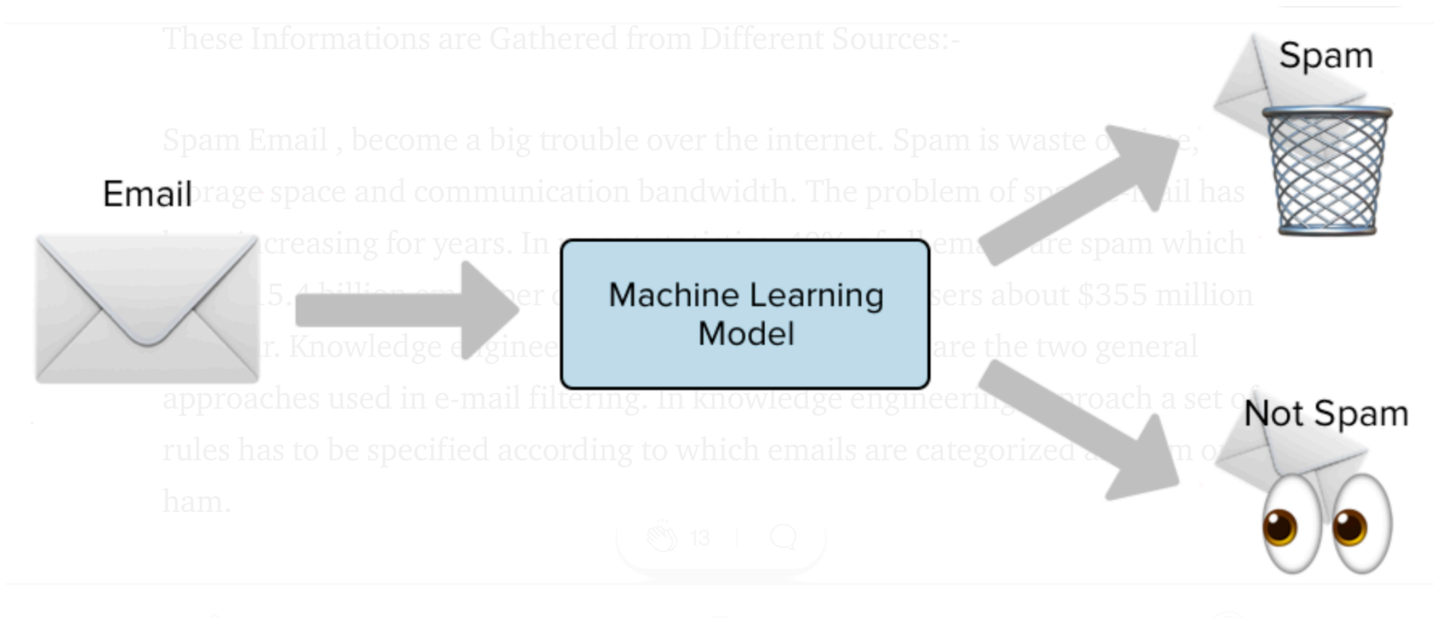
- ML lets agents **learn how to deal with uncertain environments.**
- In fact, agents **learn to optimize their knowledge** to achieve the predefined goals.



Example: Spam email filtering

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

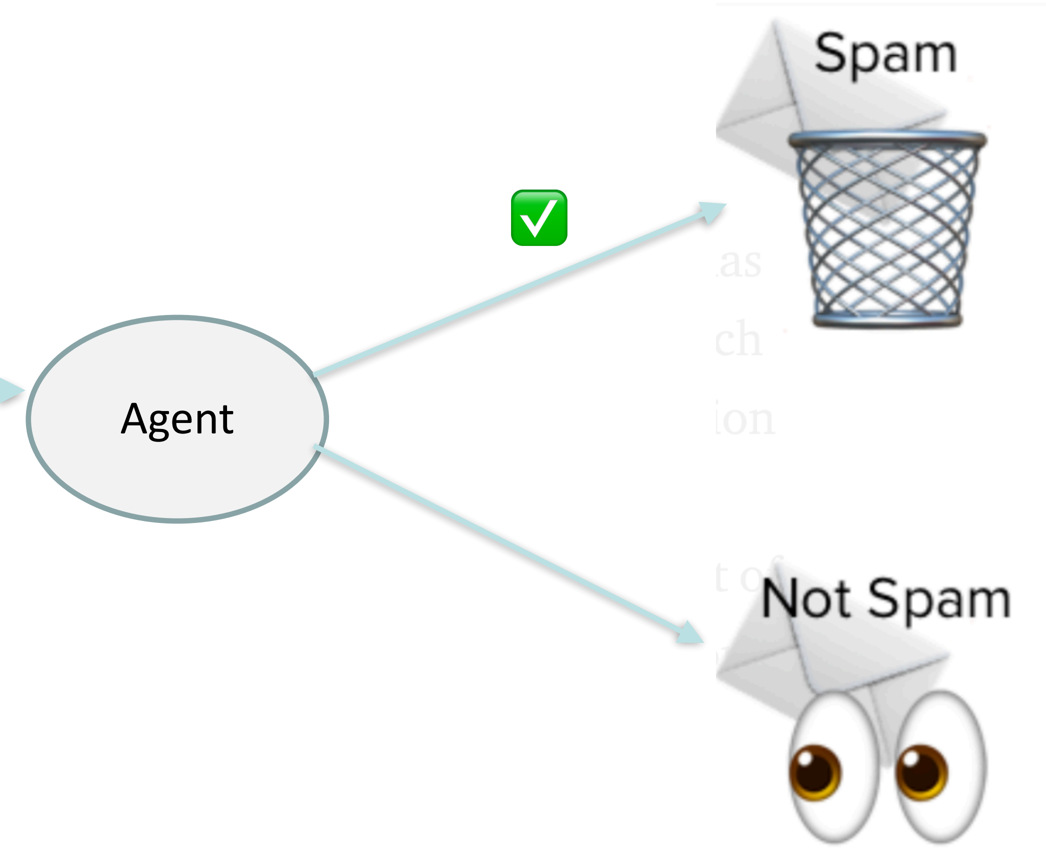
Why is this task important? Spam prevents the user from making full and good use of time, storage capacity and network bandwidth



https://medium.com/analytics-vidhya/email-spam-classifier-using-naive-bayes-a51b8c6290d4

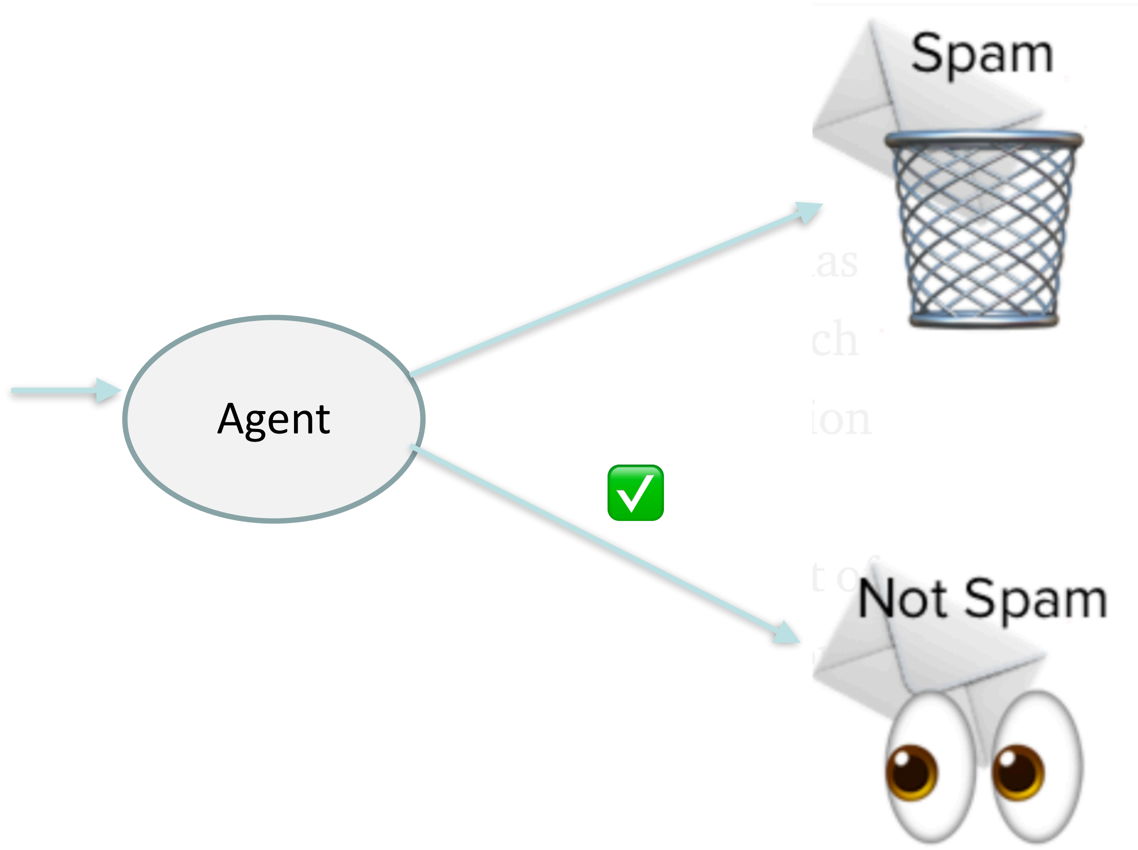
Example: Spam email filtering

Hello,
Do you want free printer cartridges? Why pay more when you can get them ABSOLUTELY FREE! Just...



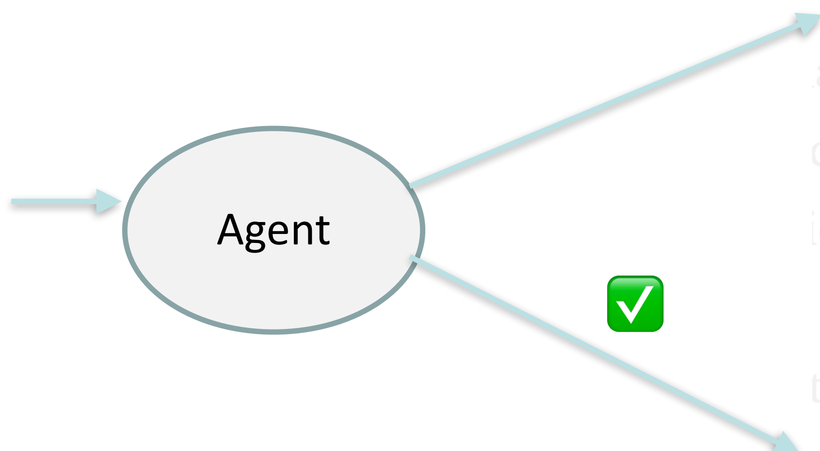
Example: Spam email filtering

Hi Anna,
how is it going in with the new apartment? I just wanted to invite you and John for my birthday party next weekend...



Example: Spam email filtering

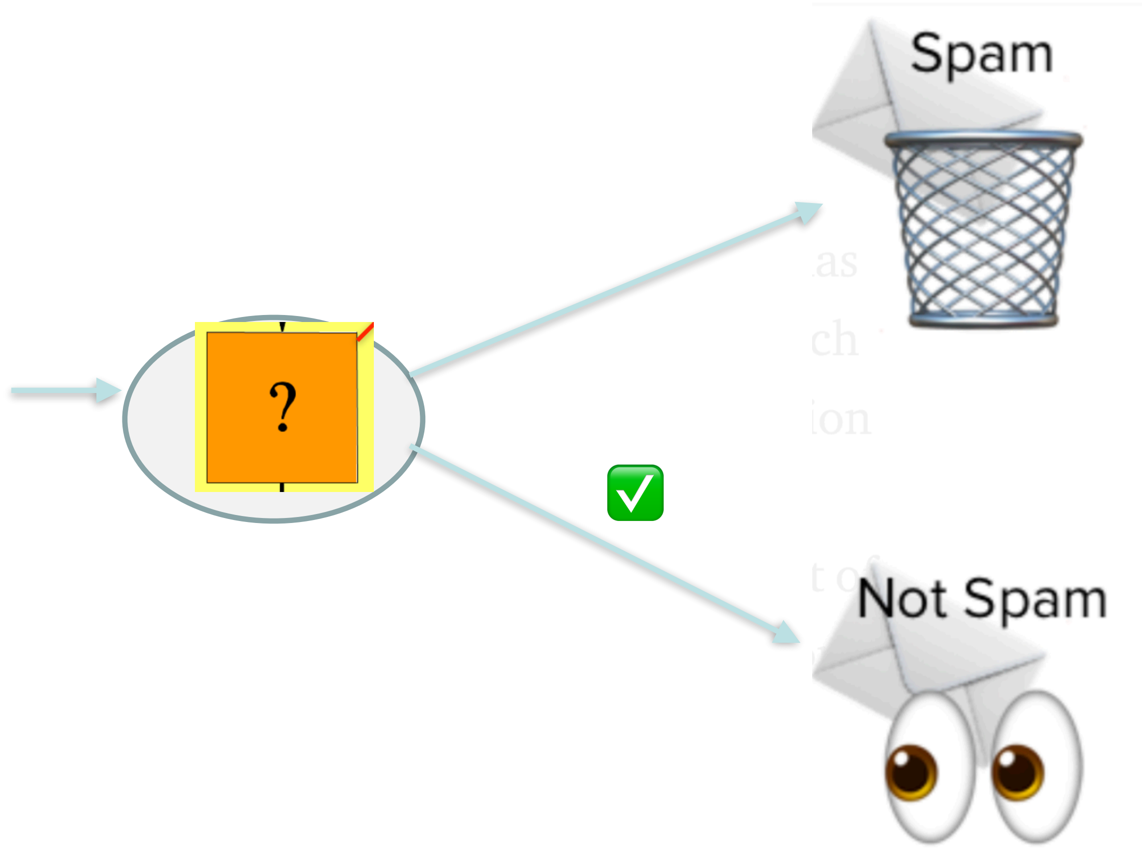
Hi Anna,
how is it going in with the new apartment? I just wanted to invite you and John for my birthday party next weekend...



Why is the agent uncertain about which action to choose?

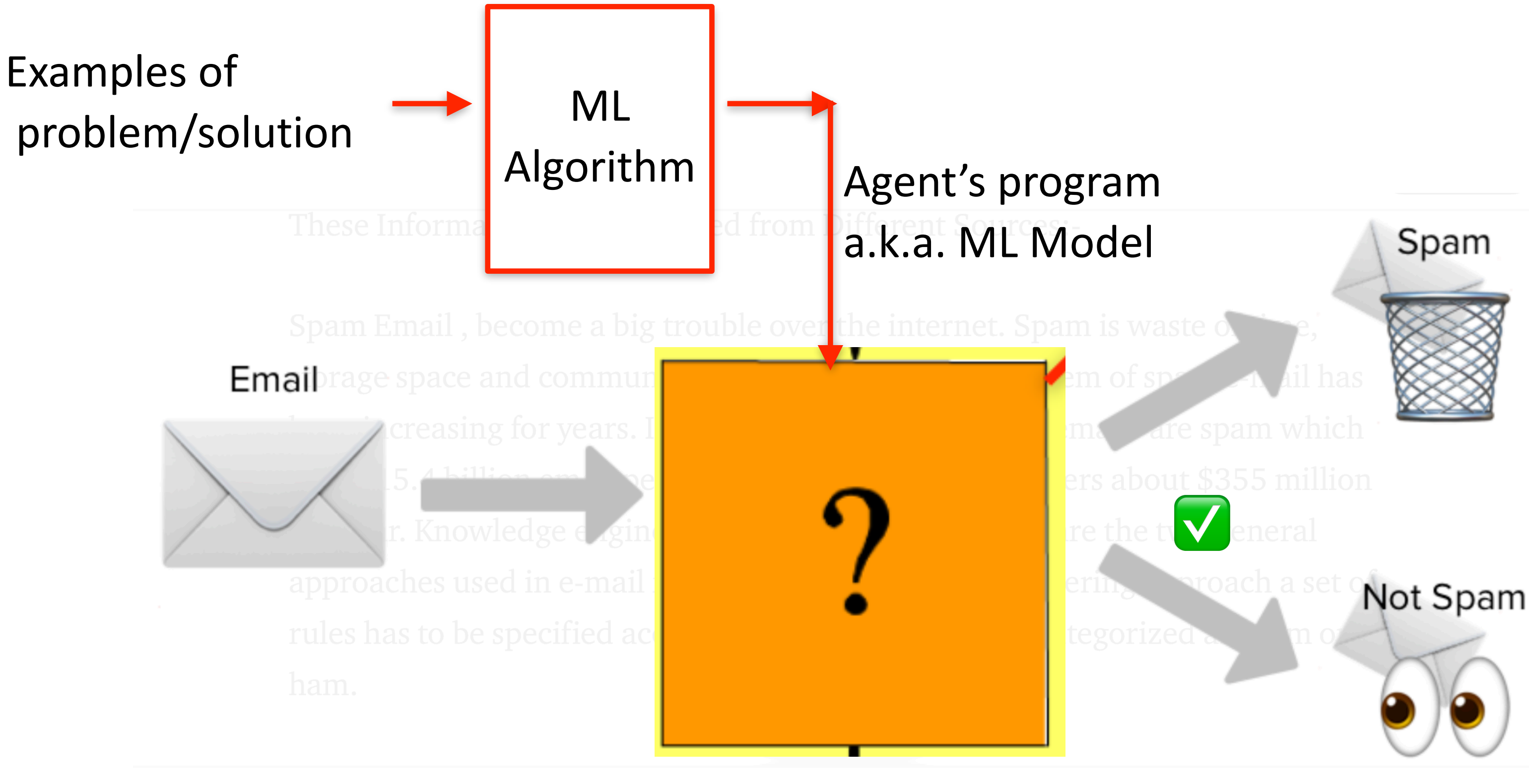
Spam email filtering: agent

Hi Anna,
how is it going in with the new apartment? I just wanted to invite you and John for my birthday party next weekend...

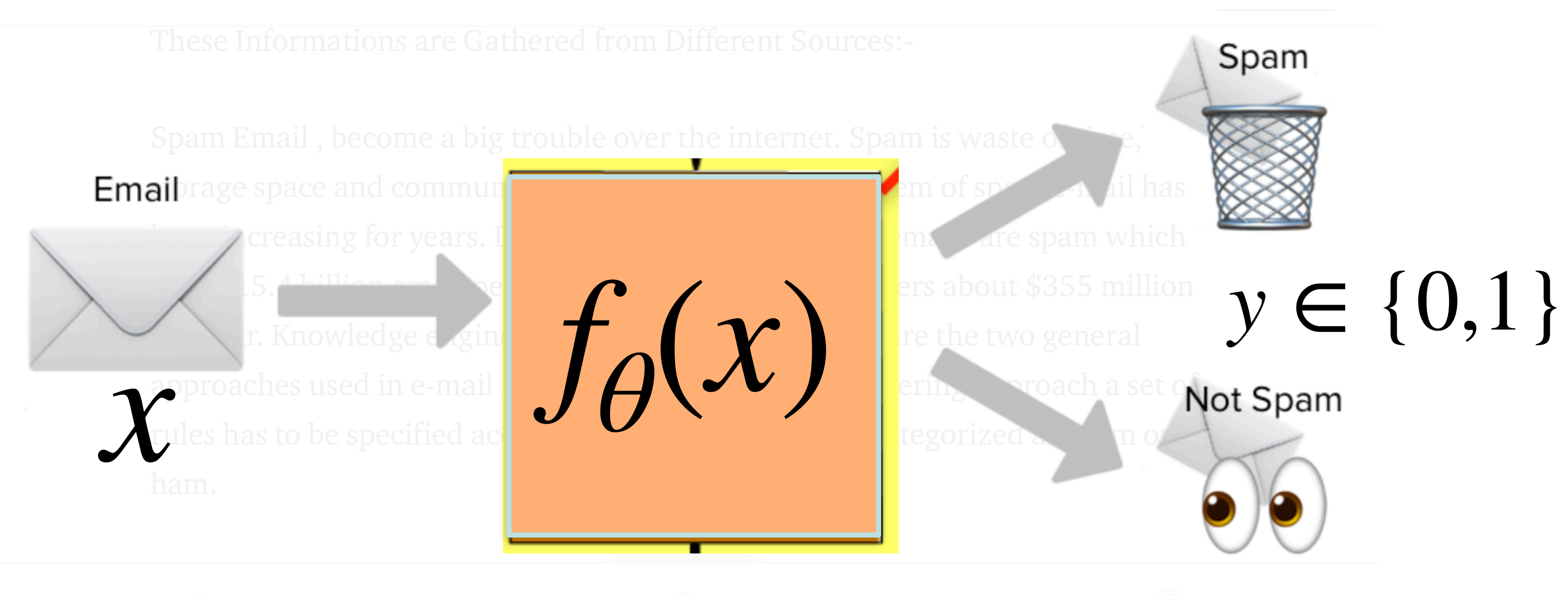


Spam email filtering: ML to define agent

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken



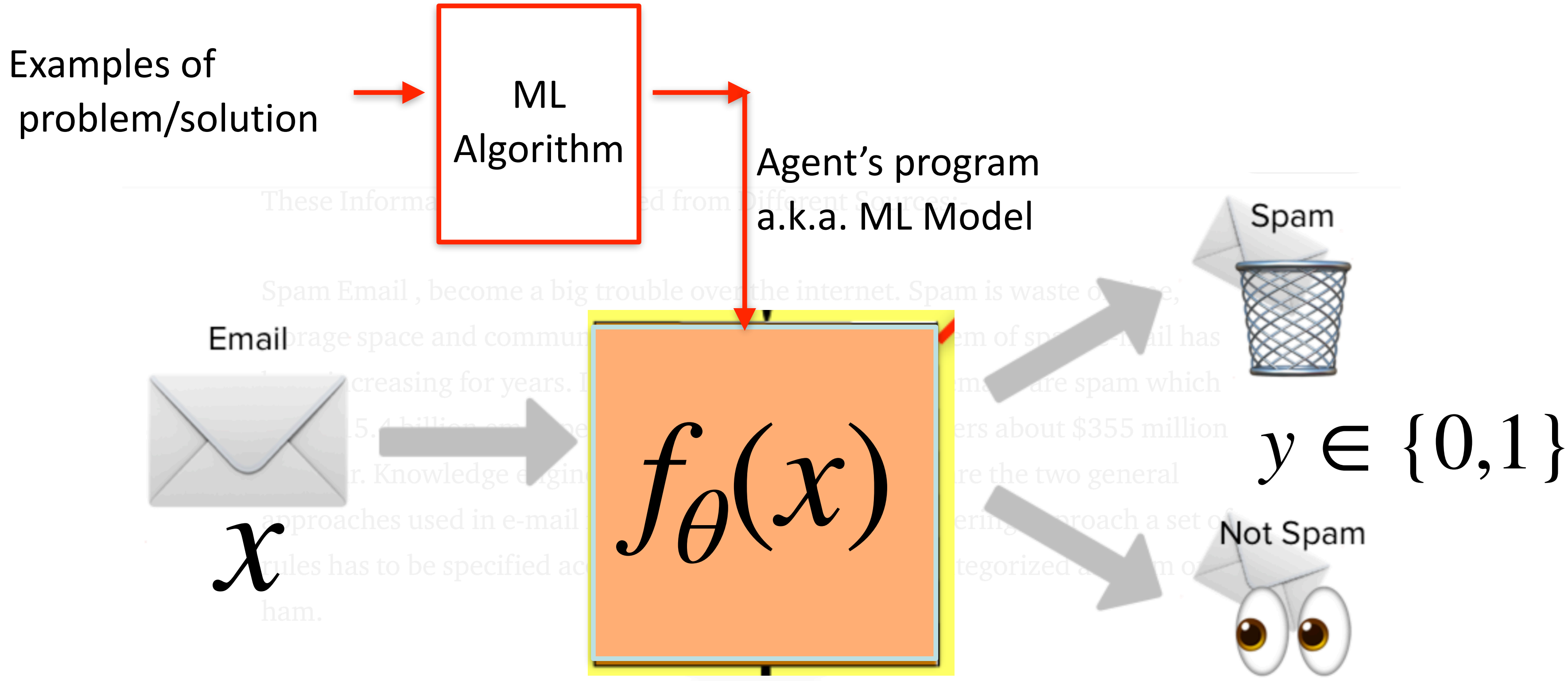
Spam email filtering: Inference phase



X, and y are variables (they can have different values).

θ indicates a set of parameters that define function f

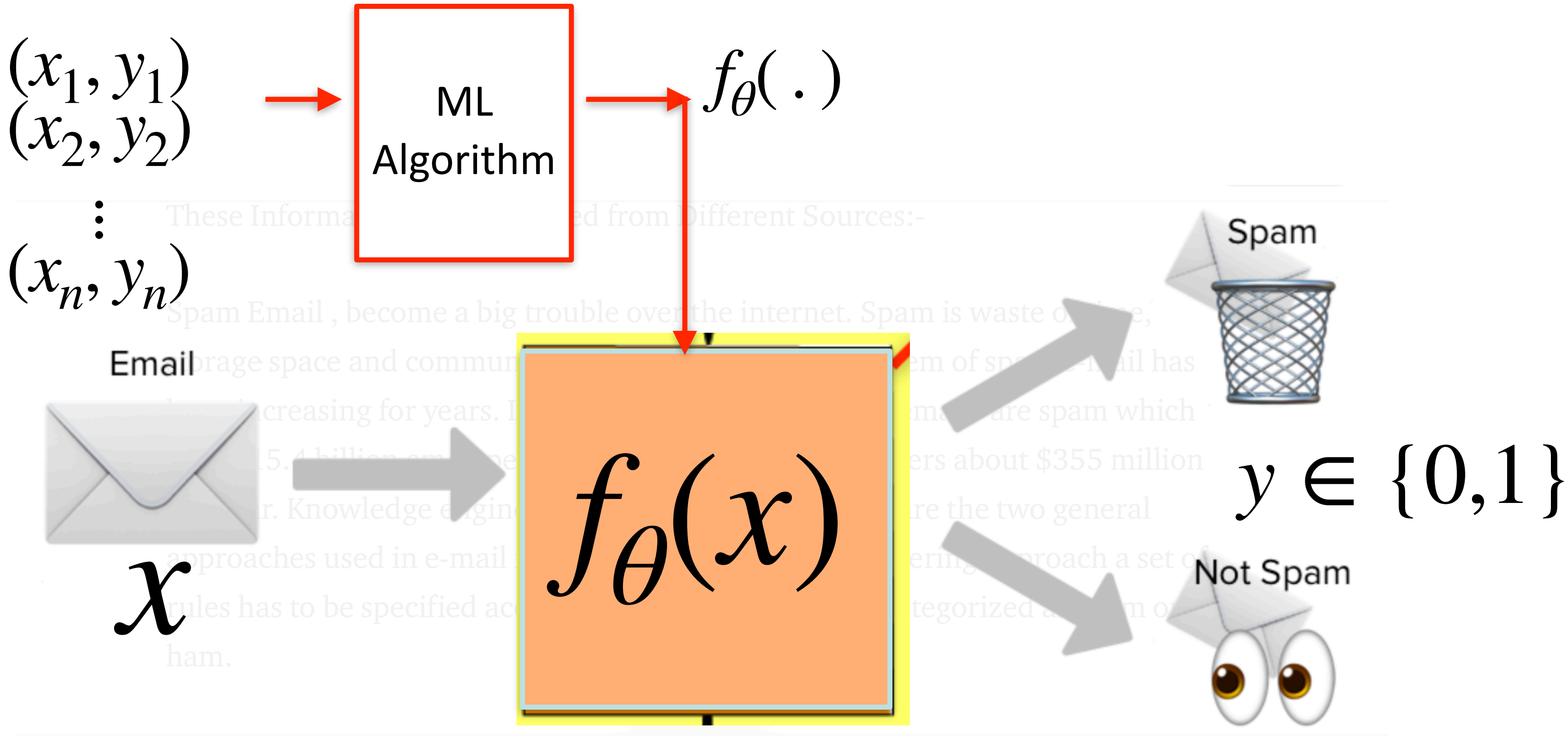
Spam email filtering: Learning phase



Spam email filtering: Learning phase

UNIVERSITÄT DUISBURG ESSEN

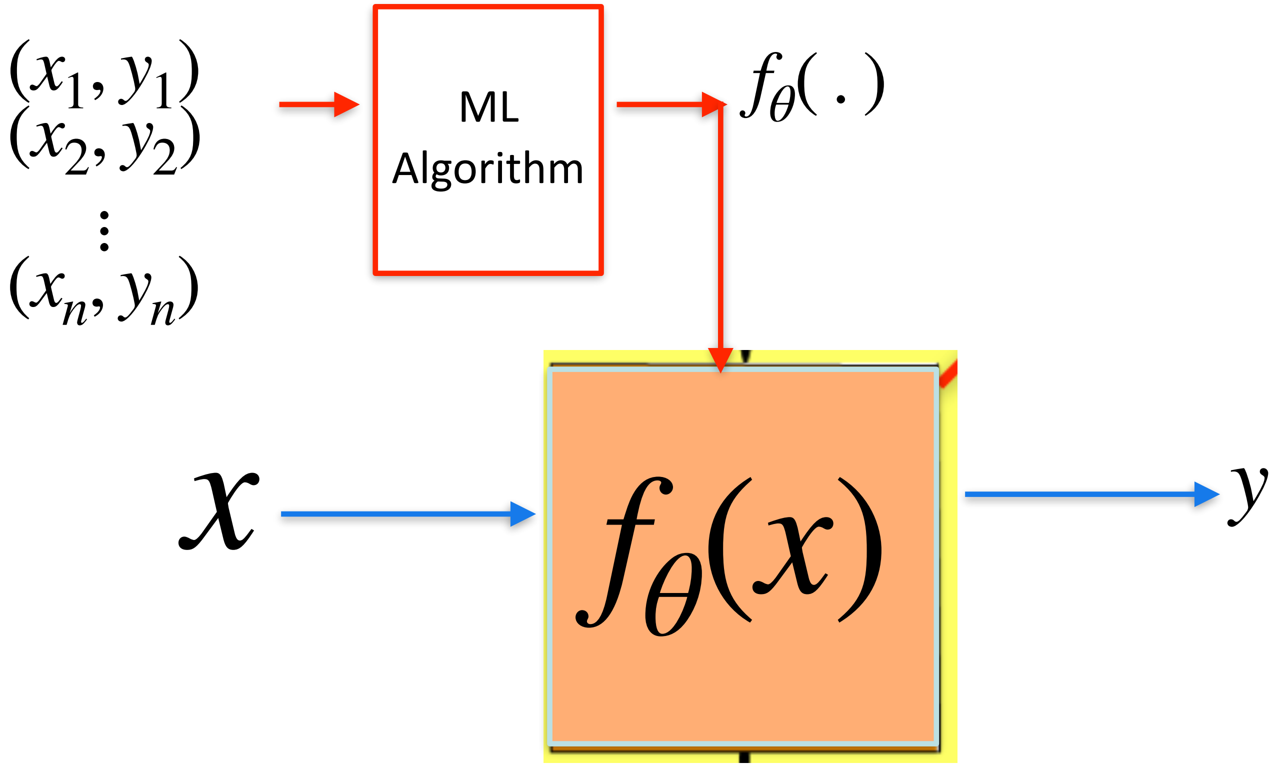
Offen im Denken



Inference

$$x \longrightarrow f_\theta(x) \longrightarrow y$$

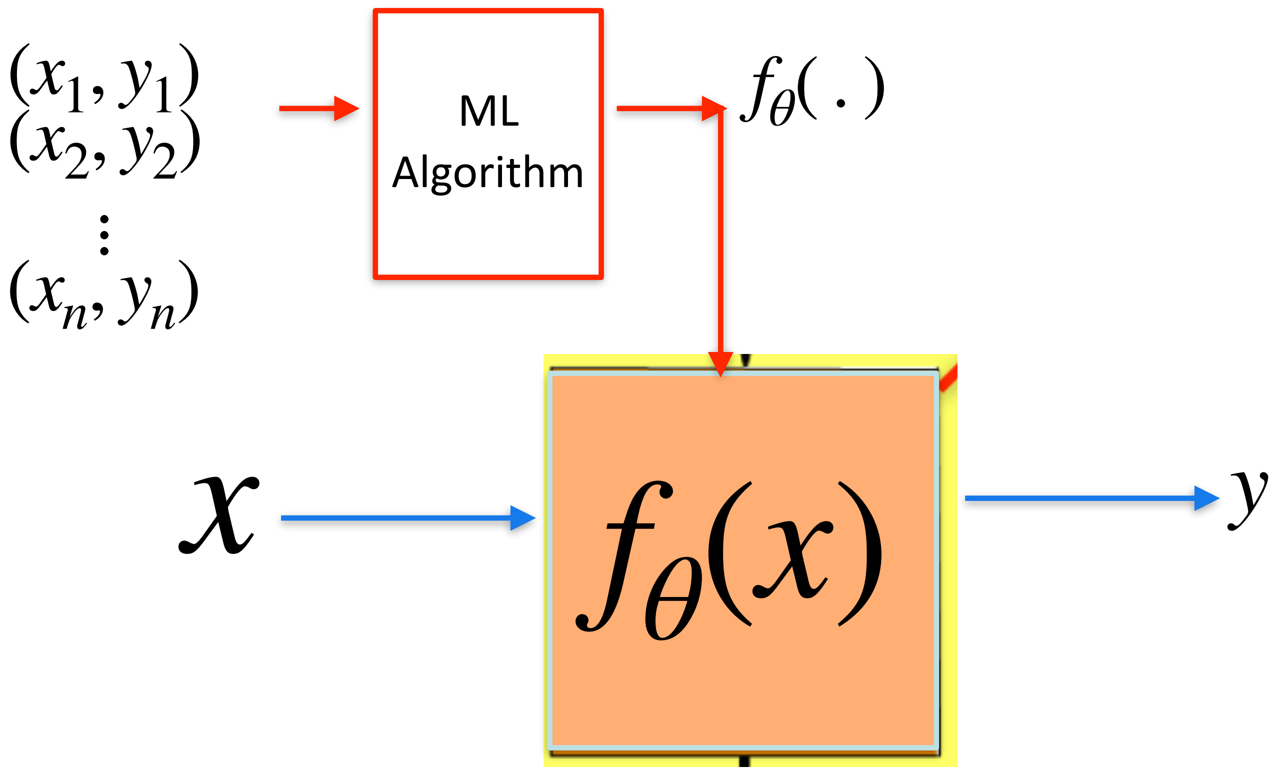
Learning



Learning

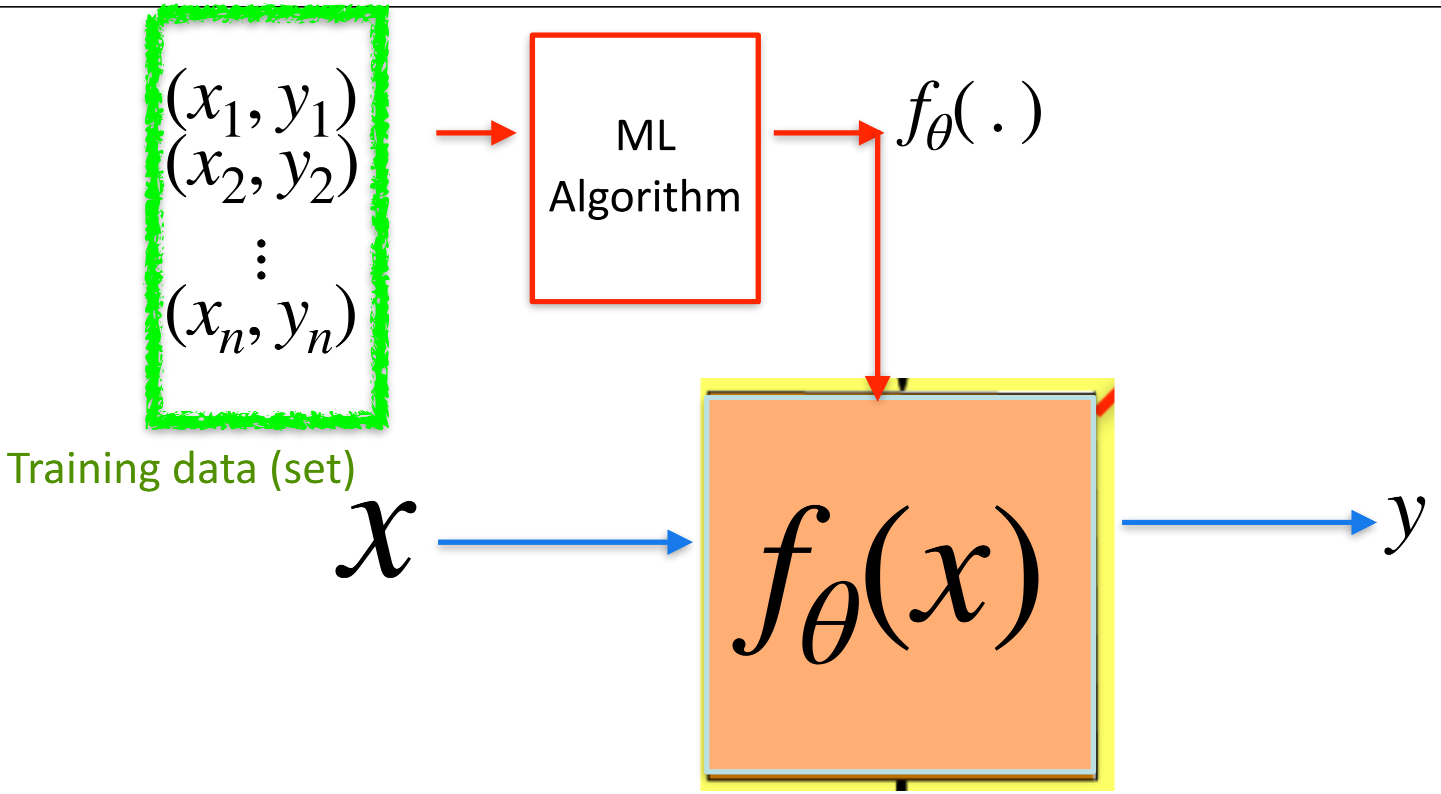
Universität Duisburg Essen

Offen im Denken



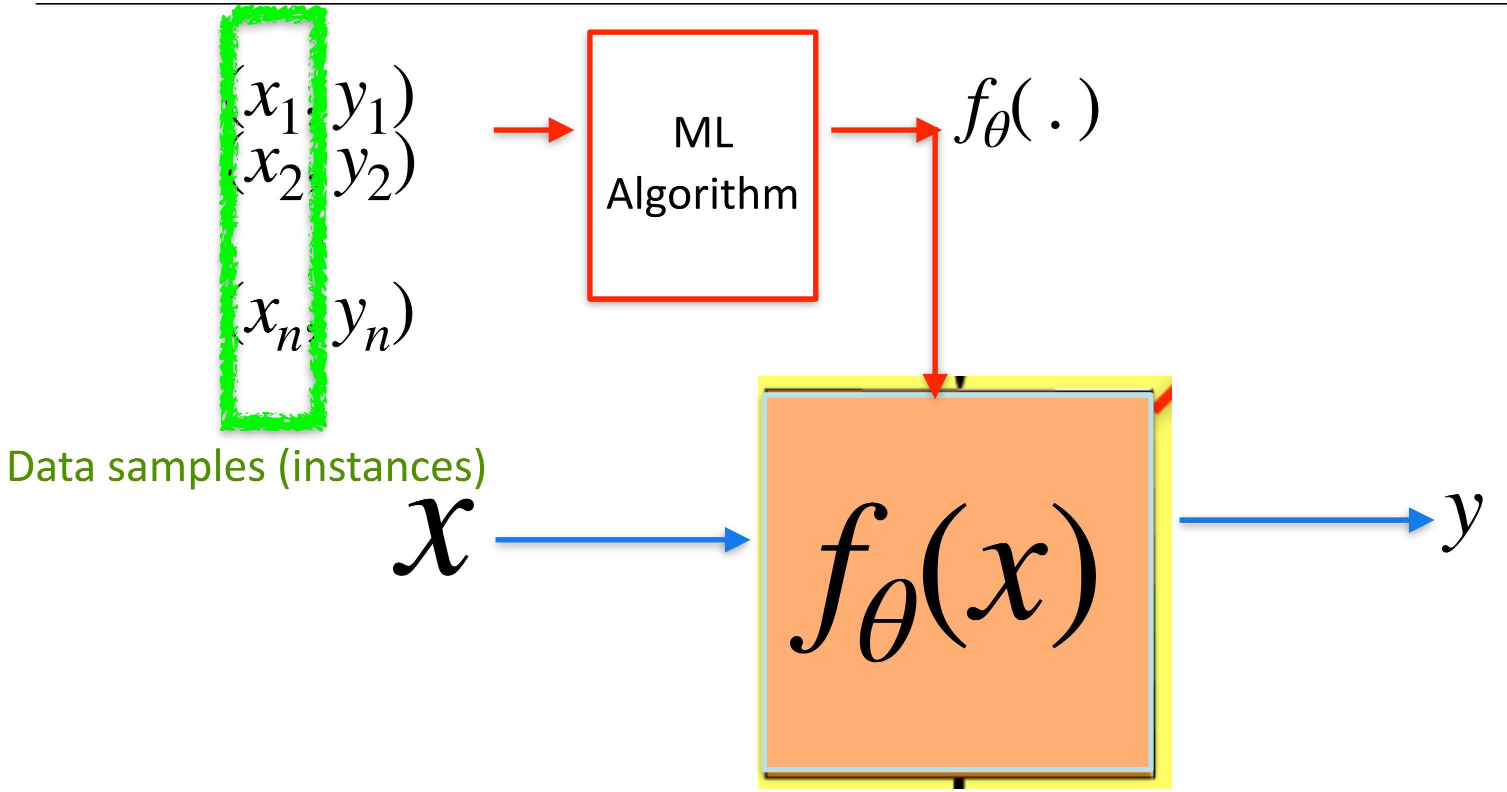
In ML, we aim to optimize parameters θ so that they capture as much as knowledge required to map input x to output y.

Terminology

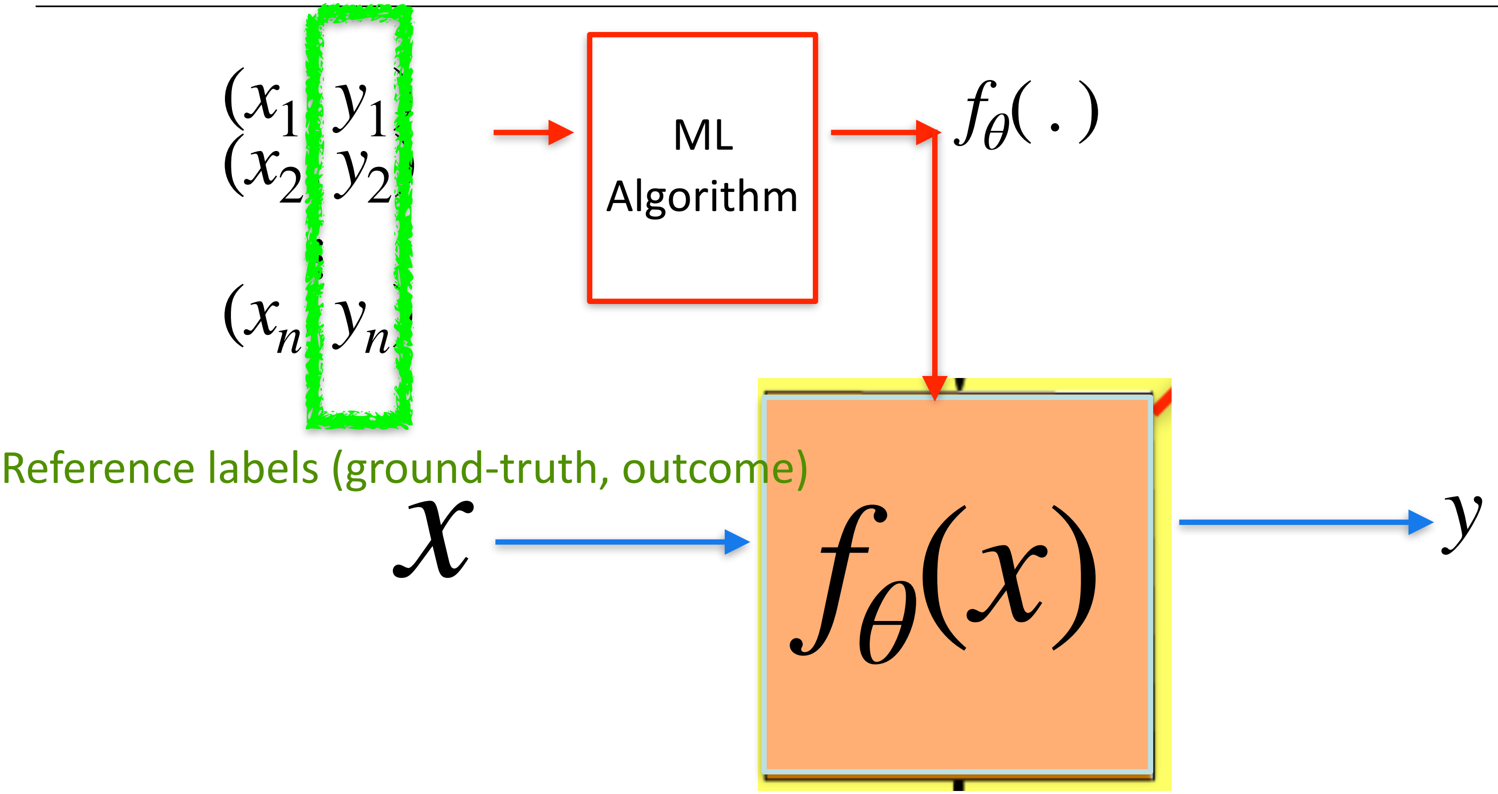


Terminology

Offen im Denken

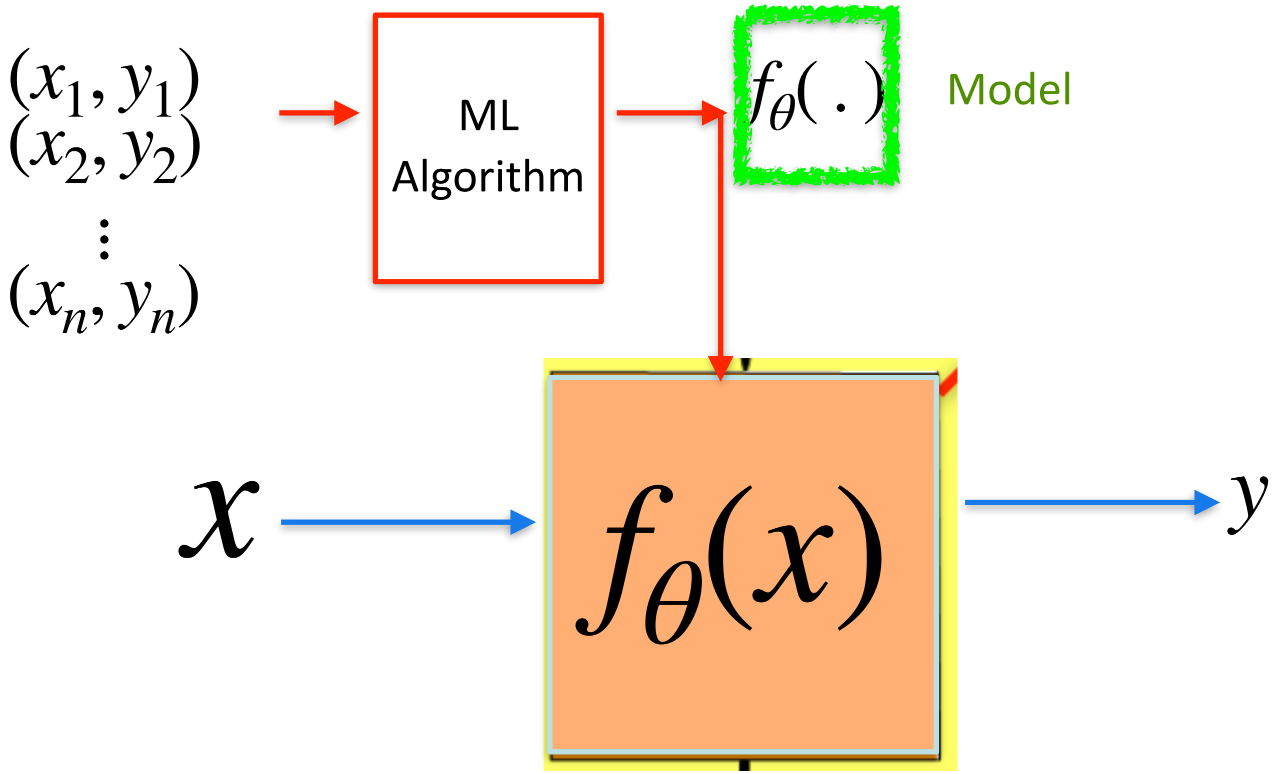


Terminology

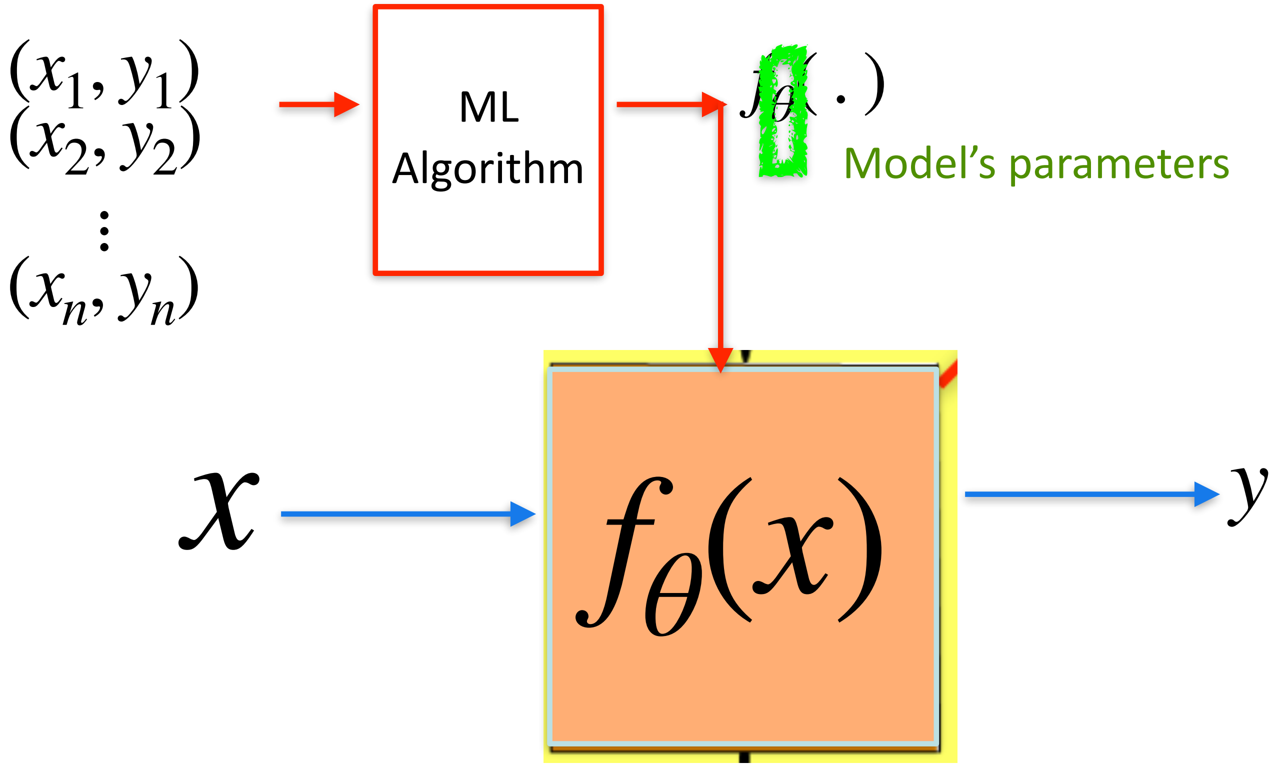


UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

Terminology



Terminology

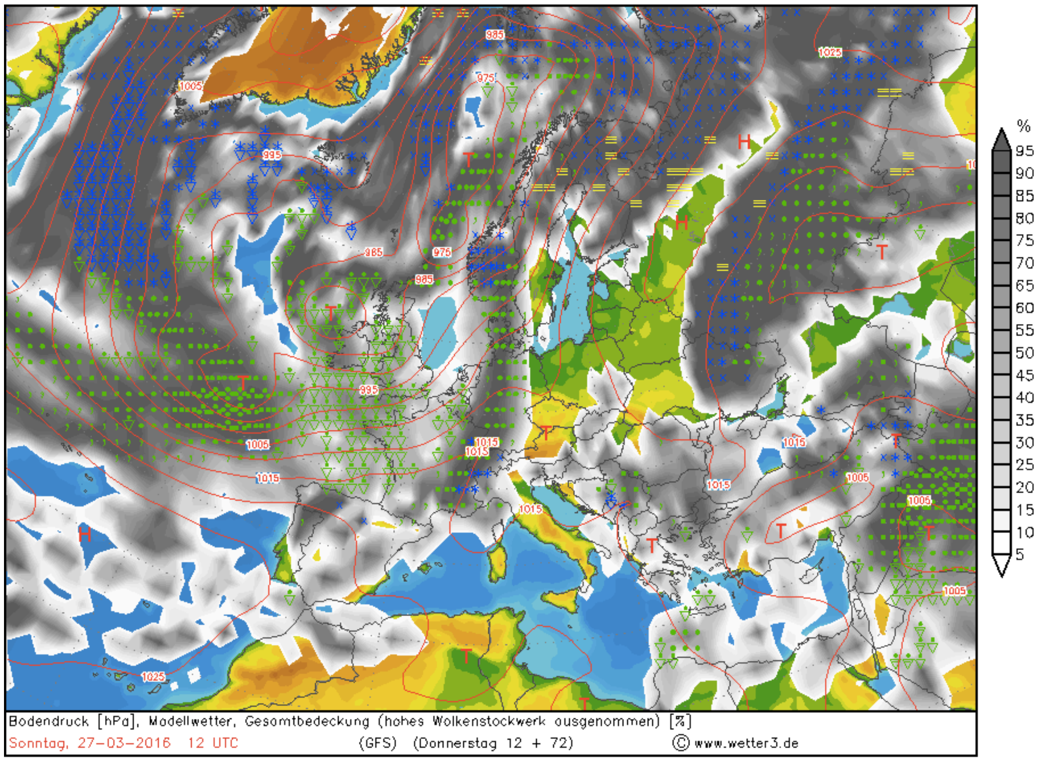


Feature

Feature

- Models need to be implemented on computers
- However, computers only understand binary language, e.g., 0110111
- How can we **encode (a.k.a represent)** instances such that computers can process them?
  - **Features**
- Features are measurable properties that can describe instances in one experiment.
- Features are better to be **informative**, **discriminating** and **independent**.

Example: Weather forecast



http://www.orniwetter.info/wp-content/uploads/2016/03/20160327_Modellwetter.gif

Possible Features

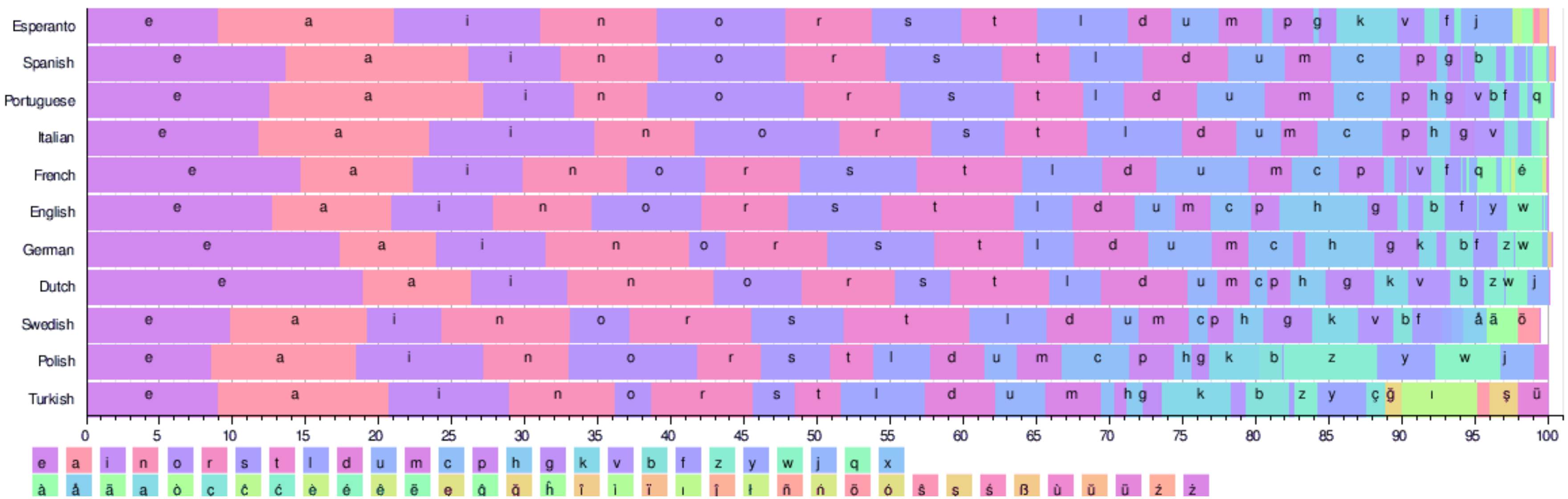
- Yesterday's weather
- Current temperature
- Current humidity
- Air pressure
- …

Possible features for language identification?

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Character set
- Special words
- Length of tokens
- Distribution of characters / character bigrams



Feature vectors

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

- Features are variables and can take values.
- We can encode the knowledge about each data sample by values of features that we defined for an experiment.
- This process is also known as **feature extraction or encoding.**
- The module that does the process is named **feature extractor** or **encoder.**

Example: Weather forecast

Possible Features

- F1: Yesterday's highest temperature
- F2: Current temperature
- F3: Did it rain yesterday?
- F4: Air pressure
- …

x_i : weather in day i

ID	F1	F2	F3
1	19	21	Yes
2	21	23	No
3	23	27	No
4	27	26	No

Example: Weather forecast

Possible Features

- F1: Yesterday's highest temperature
- F2: Current temperature
- F3: Did it rain yesterday?
- …

x_i : weather in day i

Feature vector that Represent x_1

ID	F1	F2	F3
1	19	21	Yes
2	21	23	No
3	23	27	No
4	27	26	No

Missing values

- Missing values for features are a critical issue for many algorithms

→ Replace missing values with mean or median of feature.

→ Apply smoothing algorithm. Many different smoothing algorithms exist. Most simple one: Add-one-smoothing (every feature occurs at least once).

Feature types

Input (x)	feature vector	y
Hello, Do you want free printr cartriges? Why pay more when you can get them ABSOLUTELY FREE! Just	# "free": 2 YOUR_NAME: 0 # Misspelled: 2 FromFriend?: 0 ...	Spam or NotSpam
	Pixel-7,12: 1 Pixel-7,13: 0 ... # Loops: 0 ...	2

Feature types

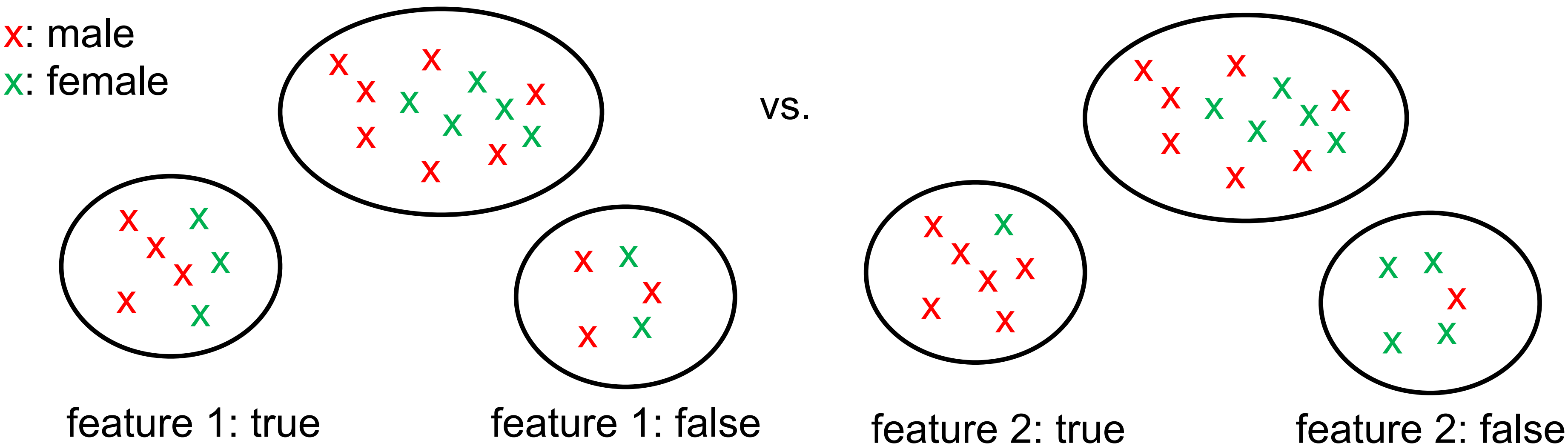
Common feature types:

- **nominal (categorical):**
  - “mode of transportation” —> {car, bus, train, tram, bicycle}
  - there is no agreed way to order these from highest to lowest.
- **binary:**
  - “isEngineer” —> {yes, no}
  - A specific type of categorical features
- **ordinal:**
  - Similar to categorical but there is a clear ordering of the categories
  - “educational experience” —> {elementary school graduate, high school graduate, some college and college graduate}
  - There is an order between values
  - distance between values is identical
- **Numerical**
  - The value is a number in a valid interval
  - “weight” $\in [0,150]$

Feature Selection

- A good feature is able to split the instances into two (or more) parts so that each part is as pure as possible
  → ideally, each part contains only examples of a single class

Entropy: measure of impurity (also: unpredictability, unorderedness, average information content, surprise)



Entropy for a feature value

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Entropy H(S): measure of impurity in a set S of instances with respect to gold labels i (gold labels are the same as ground truth labels)

$$\mathbf{Entropy}(\mathbf{S}) = -\sum_{i=1}^{n} p\left(goldlabel_i\right) * \ logp\left(goldlabel_i\right)$$

Labels = {female, male}, S_1 = {has-long-hair=true}

$$\mathrm{Entropy}(S_1) = -\mathrm{p}(\mathrm{female}) * \log \mathrm{p}(\mathrm{female}) - \mathrm{p}(\mathrm{male}) \ * \log \mathrm{p}(\mathrm{male})$$

$$= \ -\frac{15}{18} * \log\frac{15}{18} \ - \frac{3}{18} * \log\frac{3}{18} = 0.65$$

has-long-hair	TRUE	FALSE
Female	15	5
Male	3	17

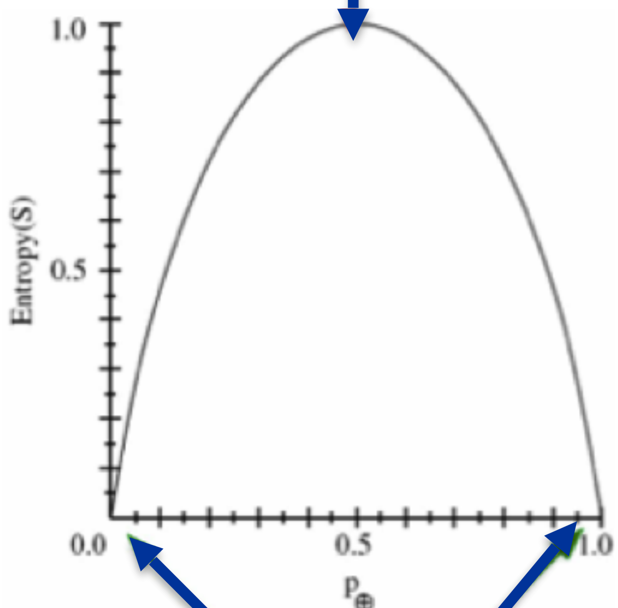
S_2= {has-long-hair=false}

$$\mathrm{Entropy}(S_1) = -\mathrm{p}(\mathrm{female}) * \log \mathrm{p}(\mathrm{female}) - \mathrm{p}(\mathrm{male}) \ * \log \mathrm{p}(\mathrm{male})$$

$$= \ -\frac{5}{22} * \log\frac{5}{22} \ - \frac{17}{22} * \log\frac{17}{22} = 0.77$$

Entropy: interpretation

Maximal value if labels are equally distributed in S



Minimal value if only one label left in S

Information of a feature

- Information = weighted average entropy over all feature values $\mathbf{k}$

$$\mathbf{I(feature)} = \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathbf{w_k * Entropy(S_k)}$$

weighted by size of part: $\mathbf{w_k} = \dfrac{|\mathbf{S_k}|}{|\mathbf{S}|}$

has-long-hair	TRUE	FALSE
Female	15	5
Male	3	17

$$\text{Entropy(has-long-hair=true)} = 0.65$$

$$\text{Entropy(has-long-hair=false)} = 0.77$$

$$\text{Information(has-long-hair)} = \frac{18}{40} * 0.65 + \frac{22}{40} * 0.77 = 0.72$$

Information Gain

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- The information gain of a feature is the change in information when applying this feature.

$$\text{Information(Before)} = \mathbf{1} * \mathbf{Entropy(S)} = -\frac{\mathbf{20}}{\mathbf{40}} * \mathbf{log}\frac{\mathbf{20}}{\mathbf{40}} - \frac{\mathbf{20}}{\mathbf{40}} * \boldsymbol{log}\frac{\mathbf{20}}{\mathbf{40}} = 1$$

$$\text{Information(Feature)} = \mathbf{0.72}$$

$$\text{InformationGain(Feature)} = 1 - 0.72 = 0.28$$

has-long-hair	TRUE	FALSE
Female	15	5
Male	3	17

Information Gain

- The information gain of a feature is the change in entropy when applying this feature.
- It can be used to automatically reduce the feature space to the most predictive features.

Filter-based feature selection:

- apply a filter (e.g. information gain) on features
- select only n-best
- train new machine learning model
- evaluate

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Data

Getting data

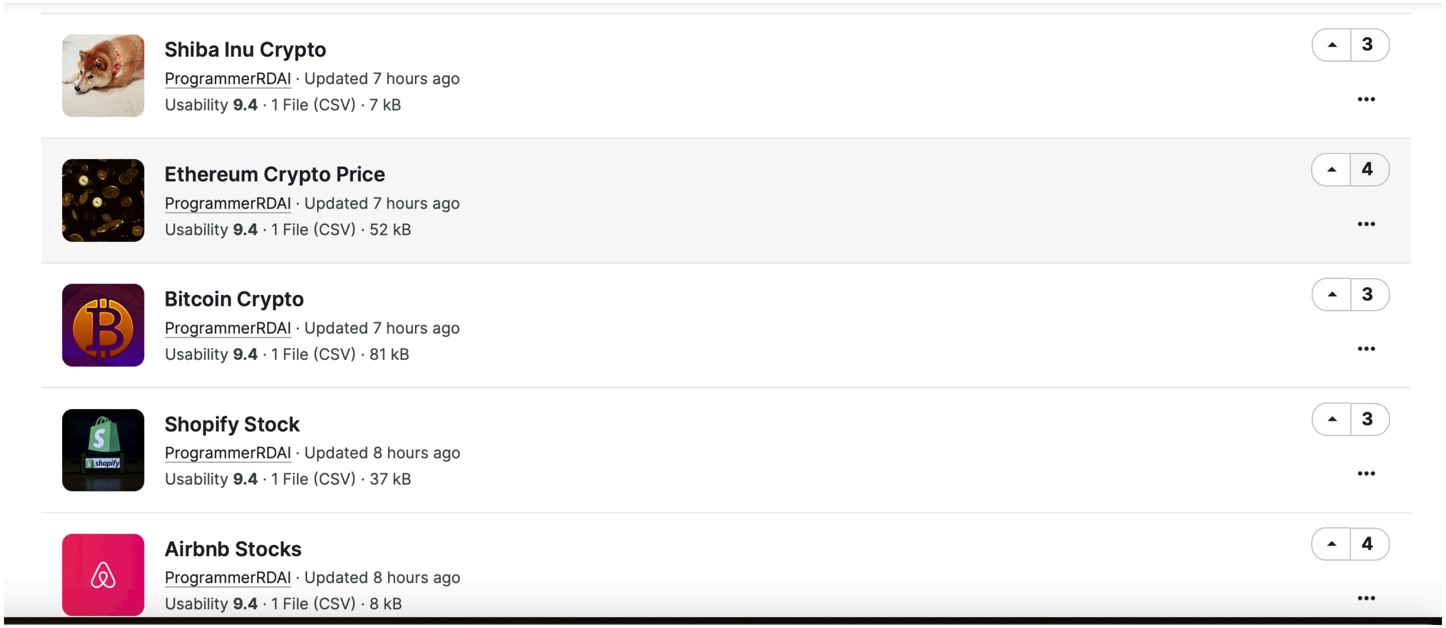
- We should collect data samples that represent the task as it may happen in deployment phase
- Our data have to contain instances to represent all possible outcome
- First try to find an existing and standard benchmark dataset (often not a perfect fit)
- We can collect data from scratch (very time-consuming)
- We sometimes adjust our formulation of the problem to get representative data

Where to get the reference labels (outcome)?

- often manually annotated by experts *(Is this email spam or not?)*
- Most ML algorithms need a considerable about of annotated data
- How much exactly depends on the problem.

Example sources to get datasets

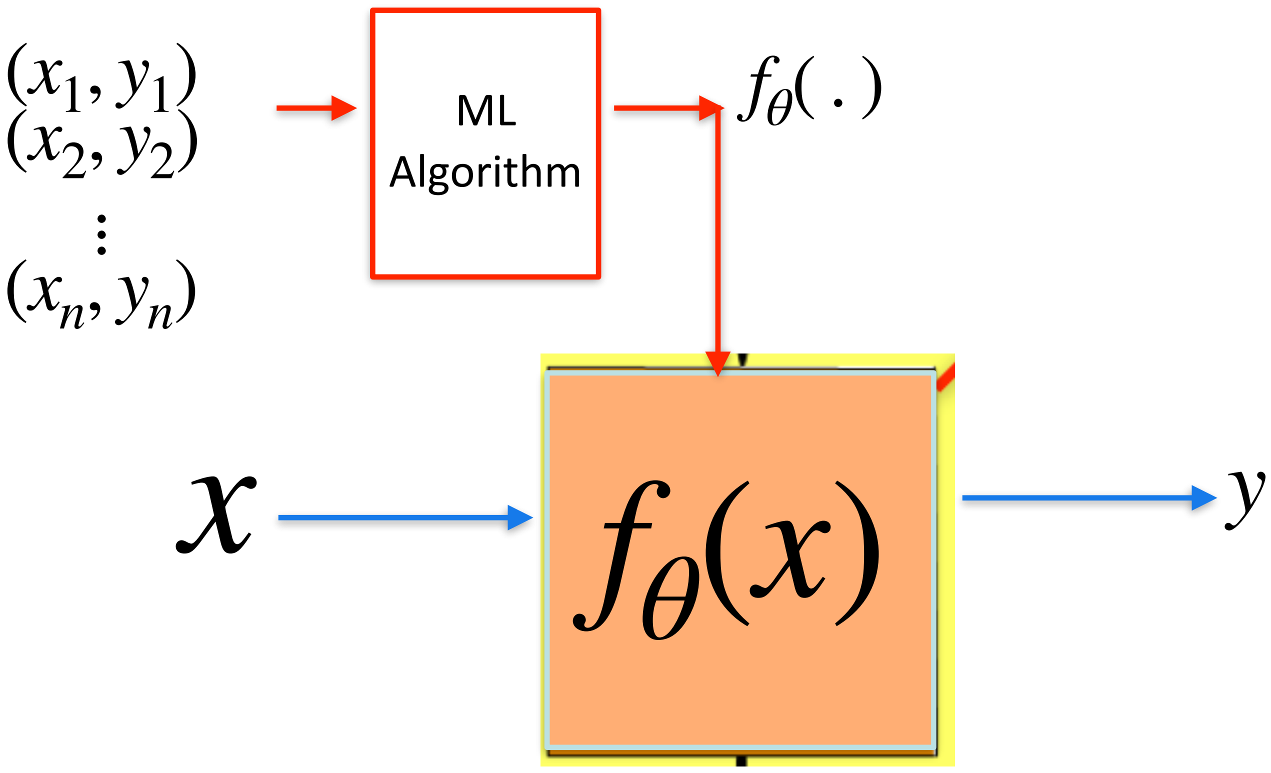
- Huggingface: (https://huggingface.co/datasets)
  - SMS_SPAM (https://huggingface.co/datasets/sms_spam/viewer/plain_text/train)
- Kaggle: (https://www.kaggle.com/datasets)



Model Training

ML

How to use data samples in a dataset to learn function $f_\theta(\,.\,)$?



Today

Supervised Learning

SL

- Training data samples x_i with reference labels y_i
- SL algorithms try to learn function $f_\theta(x)$ that approximates the data

$$x_1, y_1, \hat{y}_1 = f_\theta(x_1)$$
$$x_2, y_2, \hat{y}_2 = f_\theta(x_2)$$
$$\vdots$$
$$x_n, y_n, \hat{y}_n = f_\theta(x_n)$$

such that

The sum of the errors between $f_\theta(x_i)$ and y_i is minimum

The learned function predict y for x which has not been seen in the training data.

SL

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Training data samples $\mathrm{x_i}$ with reference labels $\mathrm{y_i}$
- SL algorithms try to learn function $f_\theta(x)$ that approximates the data

data samples

$$\boxed{\begin{array}{c} x_1 \\ x_2 \\ \\ x_n \end{array}} \begin{array}{l} y_1, \hat{y}_1 = f_\theta(x_1) \\ y_2, \hat{y}_2 = f_\theta(x_2) \\ \vdots \\ y_n, \hat{y}_n = f_\theta(x_n) \end{array}$$

such that

The sum of the errors between $f_\theta(x_i)$ and y_i is minimum

The learned function predict y for x which has not been seen in the training data.

SL

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Training data samples x_i with reference labels y_i
- SL algorithms try to learn function $f_\theta(x)$ that approximates the data

Reference outcome

$$x_1, \boxed{y_1}, \hat{y}_1 = f_\theta(x_1)$$

$$x_2, y_2, \hat{y}_2 = f_\theta(x_2)$$

$$\vdots$$

$$x_n, y_n, \hat{y}_n = f_\theta(x_n)$$

such that

The sum of the errors between $f_\theta(x_i)$ and y_i is minimum

The learned function predict y for x which has not been seen in the training data.

SL

Universität Duisburg Essen
Offen im Denken

- Training data samples x_{i} with reference labels y_{i}
- SL algorithms try to learn function $f_\theta(x)$ that approximates the data

$$
\begin{aligned}
&x_1, y_1, \boxed{\hat{y}_1 = f_\theta(x_1)} \\
&x_2, y_2, \hat{y}_2 = f_\theta(x_2) \\
&\vdots \\
&x_n, y_n, \hat{y}_n = f_\theta(x_n)
\end{aligned}
$$

Model's predictions

such that

The sum of the errors between $f_\theta(x_i)$ and y_i is minimum

The learned function predict y for x which has not been seen in the training data.

SL application

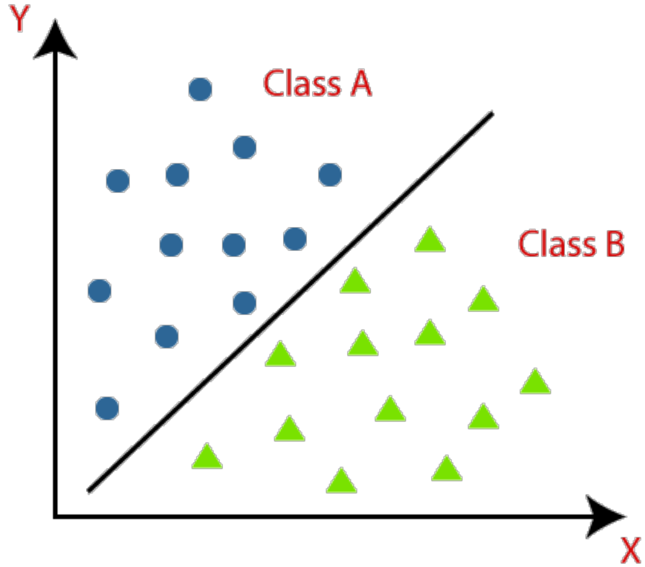
- SL has applications in building models that deal with two major categories of problems
  - **Classification**
  - **Regression**

SL application

- SL has applications in building models that deal with two major categories of problems
  - **Classification**
  - Regression

Classification

- **Goal:** Assigning an input data sample x into a category
    - y is a categorical variable (or feature)



Classification

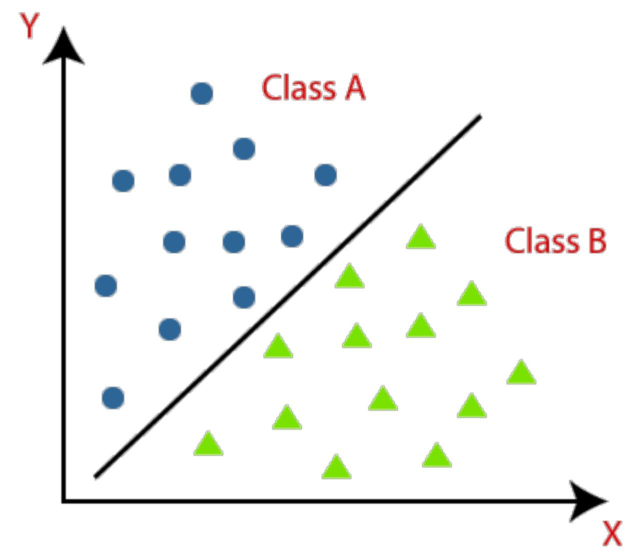
- **Goal:** Assigning an input data sample x into a category
- In these problems, y is a categorical variable (or feature)

- Different types of classification problems:
  - **Binary classification:** predict one of two classes
  - **Multi-class classification:** predict one of more than two classes
  - **Multi-label classification:** predict one or more classes for each example
  - **Imbalanced classification:** classification tasks where the distribution of examples across the class labels is not equal.

Binary classification

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

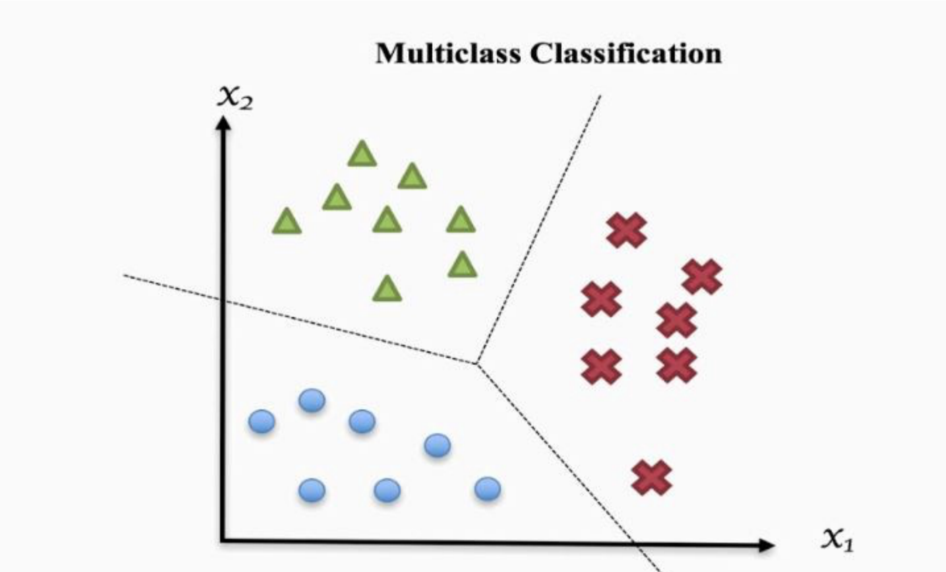
- In these problems, y is a binary variable (or feature)



- Examples:
  - Is this image a cat or dog?
  - Is this movie review positive or negative?

Multi-Class classification

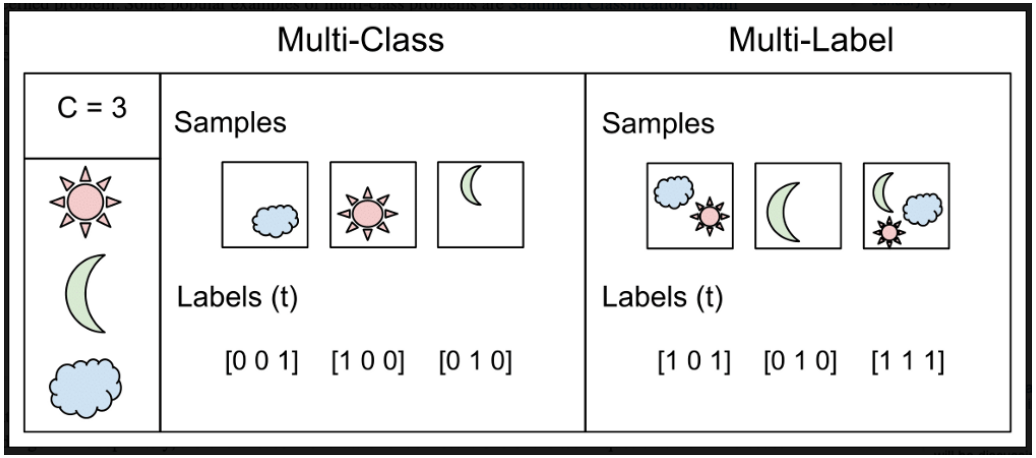
- In these problems, the number of possible labels is more than two
- **Examples**:
  - Is this image a cat or dog or panda?
  - Is this movie review very positive, positive, neutral, negative or very negative?



https://towardsdatascience.com

Multi-label classification

- In these problems, each sample can be assigned to multiple class labels



https://medium.com

- **Examples**:
  - What entities do exist in an image? (cat, dog, pandas, …)
  - What is this movie review about? (author, director, Costume Designer, …)

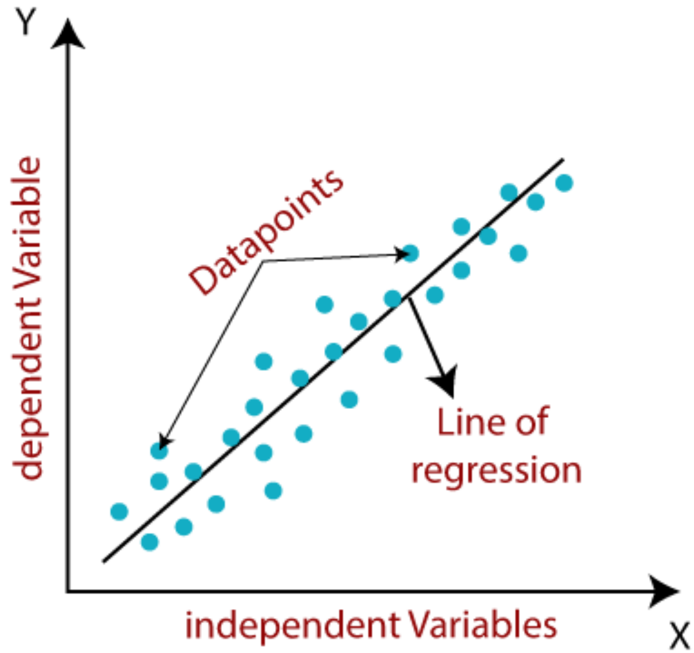
SL application

- SL has applications in building models that deal with two major categories of problems
  - Classification
  - **Regression**

Regression

Offen im Denken

- An algorithm to model the relationship between dependent and independent variables
- Some well-known algorithms
  - Linear regression
  - Polynomial regression



https://www.javatpoint.com/

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

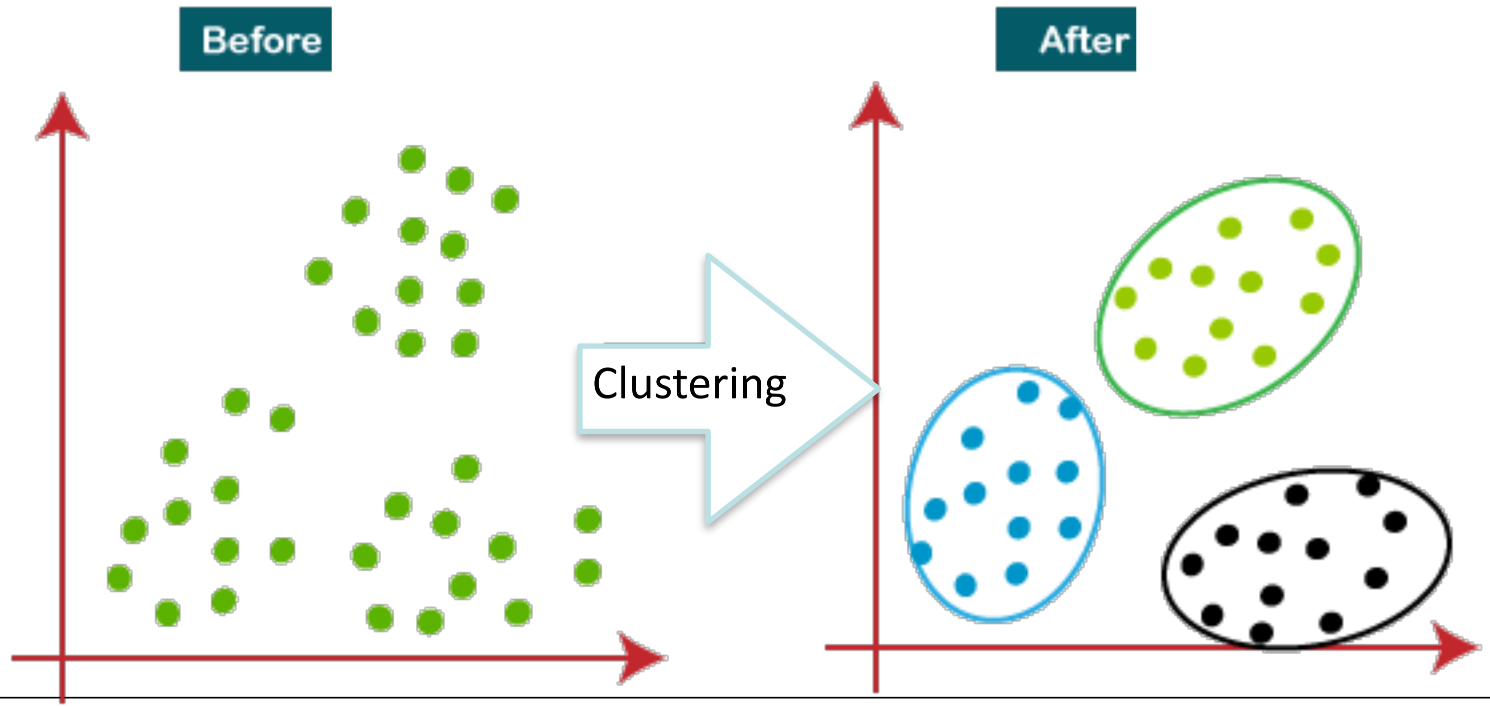
Unsupervised Learning

Unsupervised learning

- Unsupervised learning algorithms find a function that analyzes and clusters unlabeled datasets.
- These algorithms discover hidden patterns or data groupings without the need for human intervention.
- Useful for tasks that need
  - Clustering
  - Dimensionality reduction

Clustering

- Clustering algorithms group unlabeled data based on their similarities or differences
- Clustering algorithms process raw, unclassified data objects into groups represented by structures or patterns in the information.

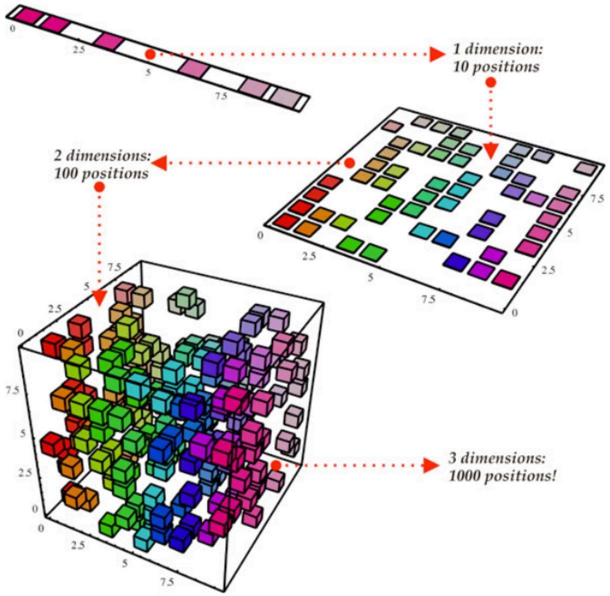


Dimensionality reduction

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Dimensionality reduction algorithms help to reduce the number of data inputs to a manageable size while also preserving the integrity of the dataset as much as possible.
- These algorithms are used when the number of features, or dimensions, in a given dataset is too high.



www.medium.com

Evaluation

How Good is my Model?

Test the model with new data.

- Provide an instance with its features, but without the label.
- *Dear model, what do you do now?*

→ The model uses the knowledge captured in its parameters to return an outcome.

Evaluate the performance of the model on new data using some examples.

What if performance not great? Improve!

- Change features
- Change classifier

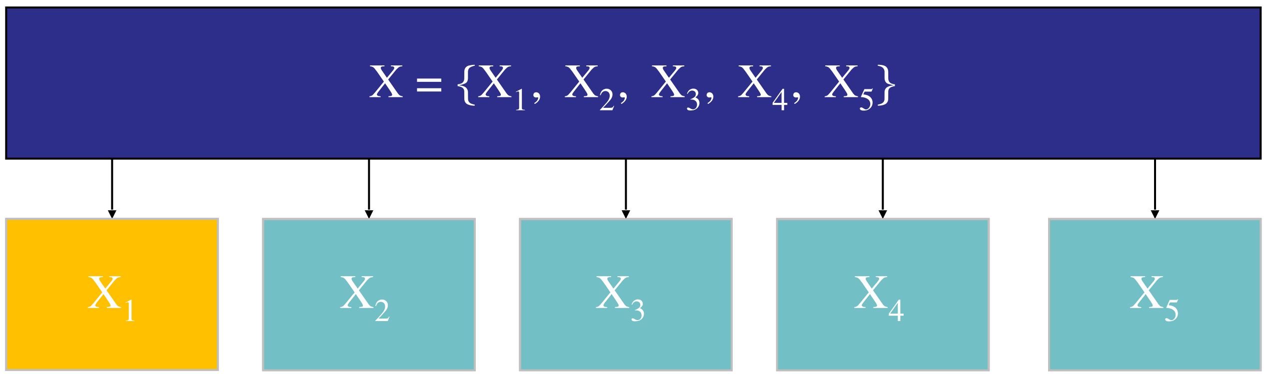
Cross-validation

- k -fold cross-validation: method to evaluate the model on the training set

1. Partition the data set into k parts of equal size.
2. Train the model on all k-1 parts and test it on the k^{th} part.
   → fold 1
3. Repeat this so that each part forms part of the test set once.
   → k folds
4. Collect the predictions for all folds and calculate the evaluation score.

- n = # of instances in training data
- If $k = n$ → leave-one-out cross-validation

5-fold cross-validation animation



Result: combine the predictions from each run and calculate the evaluation score once.

Test set

- When we have fixed all parameters of the model, we can finally evaluate on the test set.
- The test set should be representative for the data.
  - Comparable class distribution
- The test set needs to be big enough.
  - Commonly: 10% of whole data set

Accuracy Metric

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

- Accuracy $= \dfrac{\textbf{correctly classified instances}}{\textbf{all instances}} = \dfrac{\mathbf{TP + TN}}{\mathbf{TP + FP + TN + FN}}$

- Confusion matrix:

	Predicted class	
Actual Class	Class = Yes	Class = No
Class = Yes	True Positive	False Negative
Class = No	False Positive	True Negative

- In most tasks, the classes are imbalanced.
  - Spam classification: only 5% of mails are spam.
  - Classify everything as “not spam” → spam detector with 95% accuracy
  - The other way around would be even worse.

http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Class-imbalance Problem

Example: Language Identification

Scenario A	Scenario B
99 out of 100 texts are English. Accuracy of 99% when always outputting “English”.	2 out of 100 texts are English. Accuracy of 2% when always outputting “English”.

Scenario A

Scenario B

Precision, Recall, F_1-measure

Universität Duisburg Essen
Offen im Denken

	Predicted class	
Actual Class	Class = Yes	Class = No
Class = Yes	True Positive	False Negative
Class = No	False Positive	True Negative

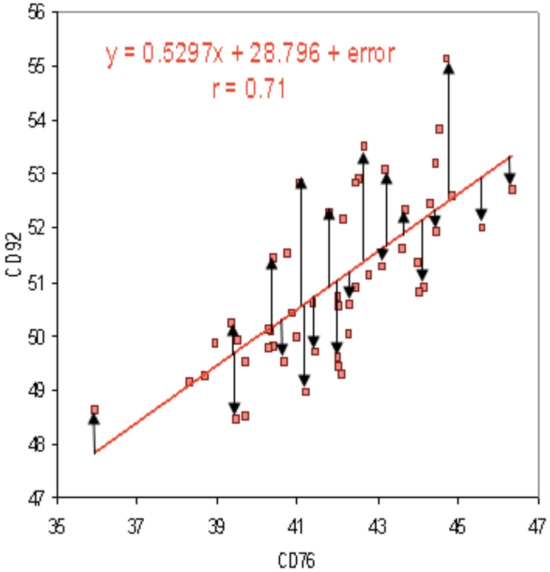
- Precision $P = \frac{TP}{TP + FP} \rightarrow$ How correct are the decisions?
- Recall $R = \frac{TP}{TP + FN} \rightarrow$ How many of the interesting cases do we catch?
- Higher recall usually leads to lower precision.
- F_1-measure $F1 = \frac{2 * P * R}{P + R} \rightarrow$ harmonic mean of precision and recall

http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

RMSE

[Root]?[Mean|Absolute][Squared]?Error

- the difference between the predicted and actual values
- requires a numeric interpretation of labels
- Algorithms (e.g. those in Weka) typically optimize these
  - possible mismatch between optimization objective and evaluation measure



Summary

- What is ML?
  - Sample representation is the core of ML
  - Feature definition and selection for sample encoding
- Two primary types of ML algorithms?
  - Supervised learning
    - Classification
    - Regression
  - Unsupervised learning
    - Clustering
    - Dimensionality reduction
- Evaluating performance of ML algorithms
  - K-fold cross validation
  - Test set
  - Evaluation metrics

Readings

Mandatory

- Russel & Norvig: *Chapter 18 Learning from Examples*, p.693-707

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Thank You