UNIVERSITÄT DUISBURG ESSEN

Open-Minded

Naive Bayes Classifier

Neuroinformatics Tutorial 3

Duc Duy Pham[1]

[1]Intelligent Systems, Faculty of Engineering,
University of Duisburg-Essen, Germany

Content

- Revision: McCulloch Pitts Neuron
- Revision Lecture
- Naive Bayes Classifier
- Tasks

Content

- Revision: McCulloch Pitts Neuron
- Revision Lecture
- Naive Bayes Classifier
- Tasks

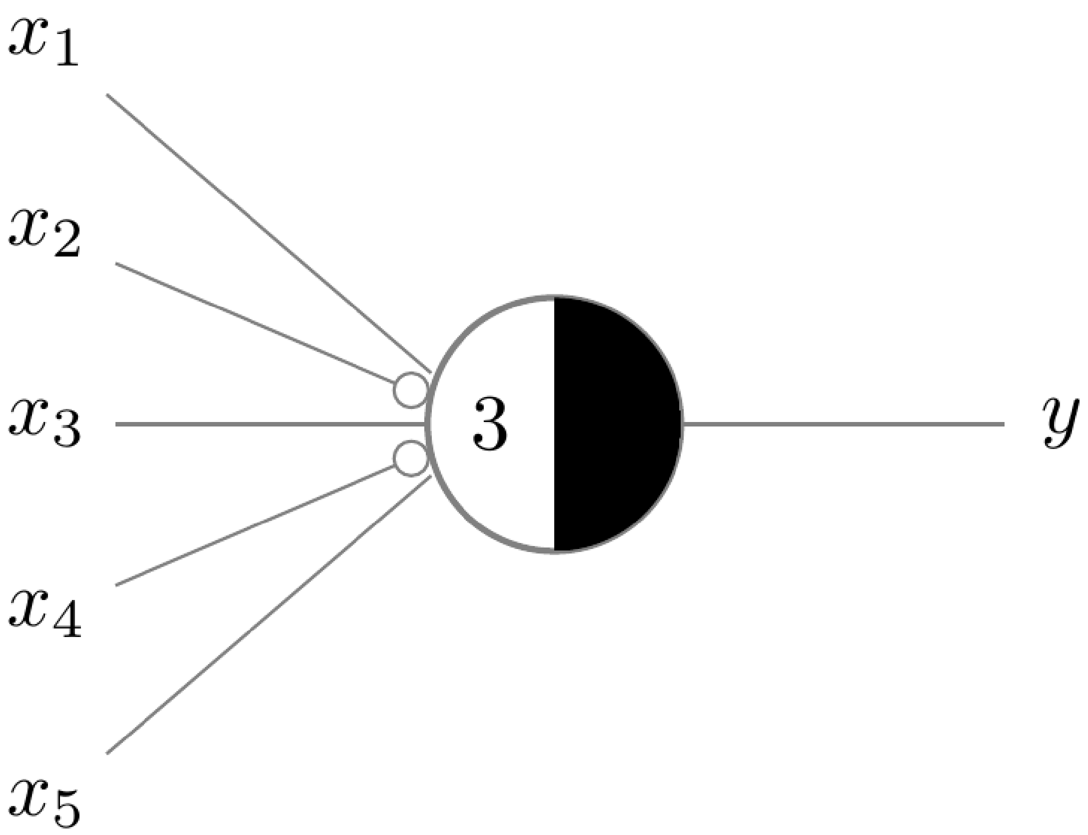
Revision: McCulloch Pitts Neuron

- Which of the following statements are true?
  1. A McCulloch Pitts Neuron can process real valued inputs
  2. A McCulloch Pitts Neuron has excitatory and activating inputs
  3. A McCulloch Pitts Neuron always returns a binary vector
  4. A McCulloch Pitts Neuron is an Artificial Neuron

Revision: McCulloch Pitts Neuron

- Which of the following parameters are crucial for the definition of a McCulloch Pitts Neuron?
  1. Threshold
  2. Number of incoming signals
  3. Number of outgoing signals
  4. Position of inhibitory and excitatory signals

The McCulloch Pitts Neuron



The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Neuron, that models NOR

The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Neuron, that models NOR

x_1	x_2	f_{NOR}
0	0	1
0	1	0
1	0	0
1	1	0

The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Neuron, that models

$$f_1(x_1, x_2, x_3, x_4, x_5) := x_1 \neg x_2 x_3 \neg x_4 x_5$$

The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Neuron, that models
$$f_2(x_1, x_2, x_3, x_4, x_5) \coloneqq \neg x_1 \neg x_2 x_3 x_4 x_5$$

The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Neuron, that models
$$f_3(x_1, x_2, x_3, x_4, x_5) \coloneqq \neg x_1 x_2 \neg x_3 x_4 \neg x_5$$

The McCulloch Pitts Neuron

- Task: Construct a McCulloch Pitts Net, that models

$$f(x_1, x_2, x_3, x_4, x_5) := f_1 \lor f_2 \lor f_3$$

McCulloch Pitts Neuron: Juypter

Content

- Revision: McCulloch Pitts Neuron
- **Revision Lecture**
- Naive Bayes Classifier
- Tasks

Revision Lecture

- Which statements are true for the learning phase?
    1. Model is trained to solve specific task
    2. Statistical properties could be used to train model
    3. This phase is also called inference phase
    4. In this phase the model gains experience

Revision Lecture

- Which statements are true for the working phase?

Revision Lecture

- According to lecture:
  - Learning Phase
  - Working Phase
  - Both phases should be applied in cycle

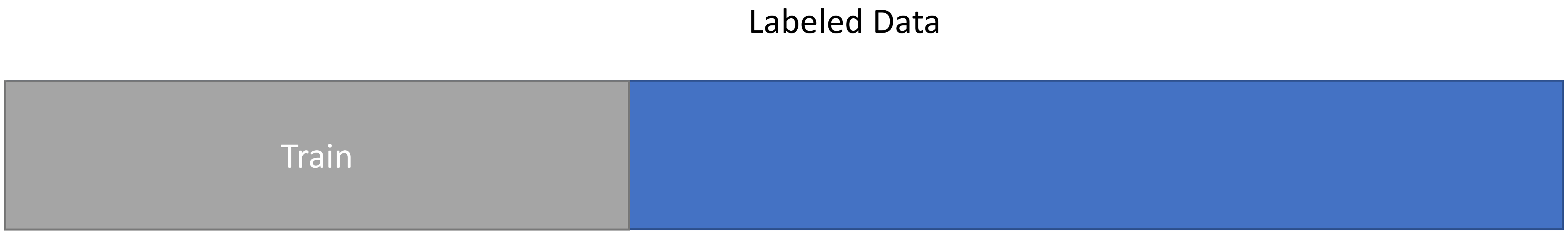
Revision Lecture

- Data partitioning:

Labeled Data

Revision Lecture

- Data partitioning:



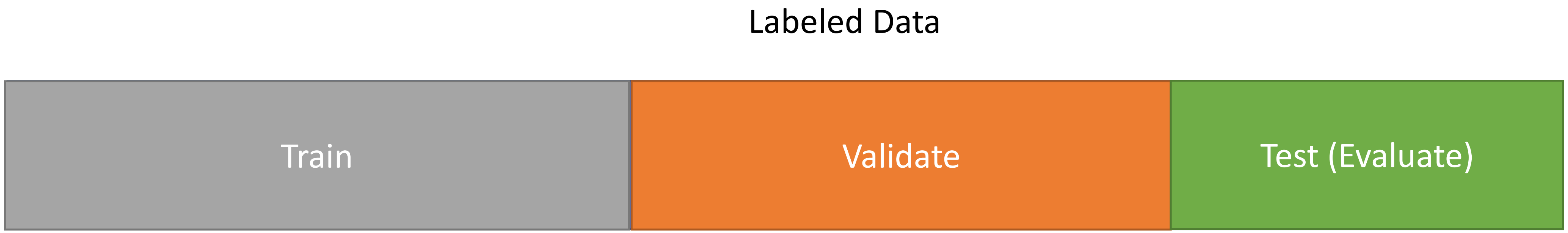
Revision Lecture

- Data partitioning:

Labeled Data

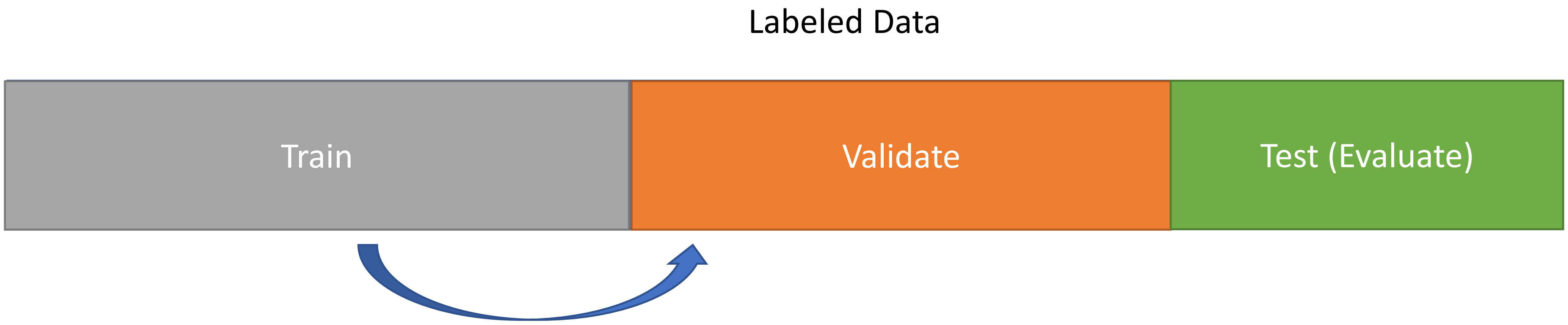
Revision Lecture

- Data partitioning:



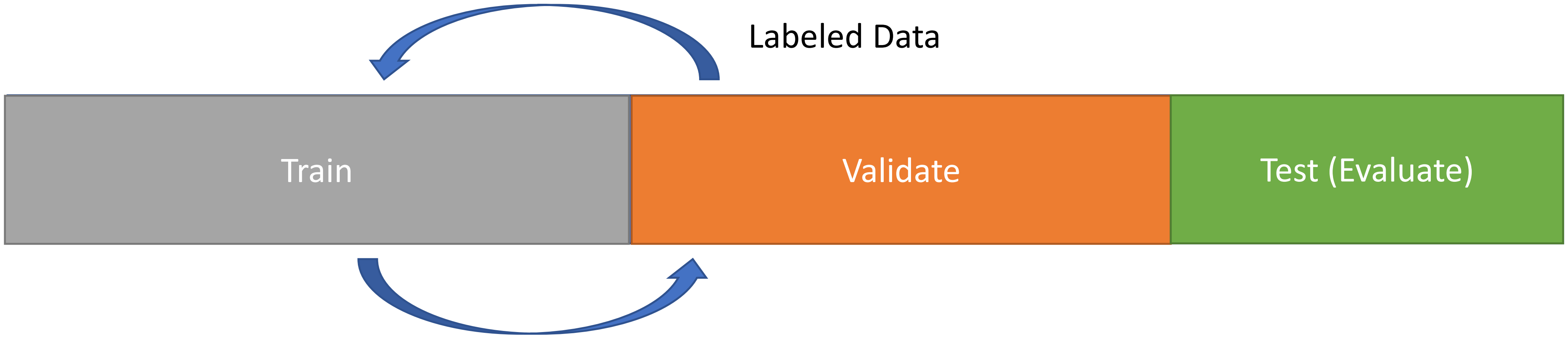
Revision Lecture

- Data partitioning:



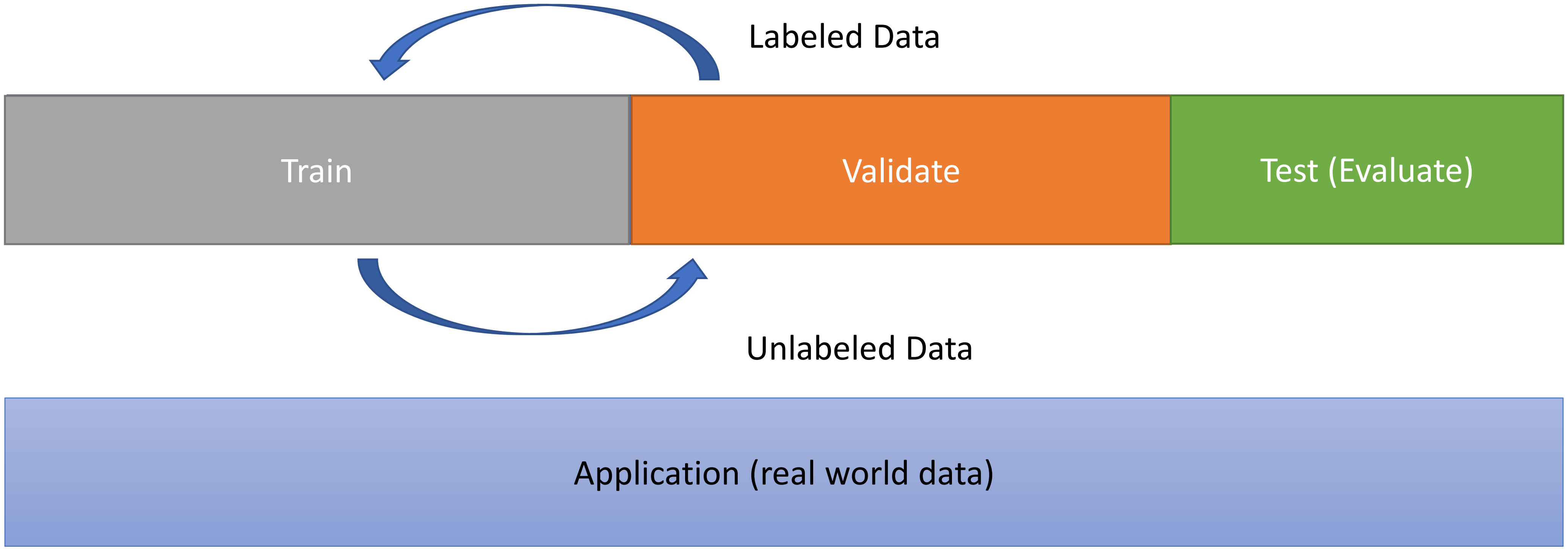
Revision Lecture

- Data partitioning:



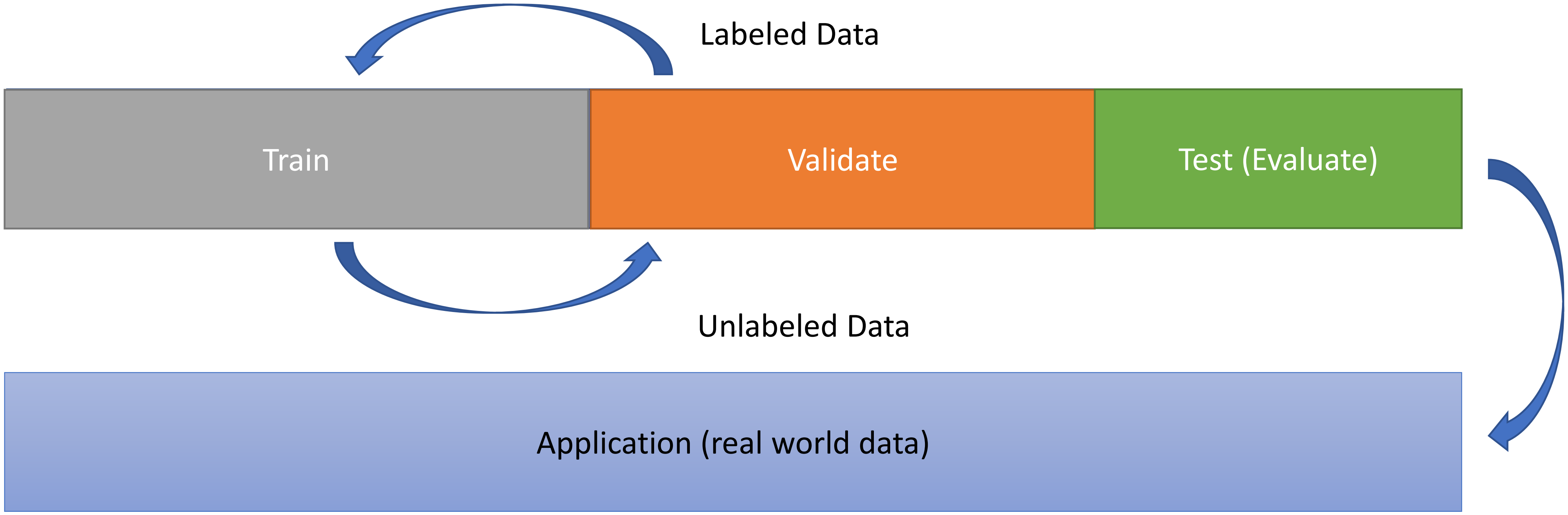
Revision Lecture

- Data partitioning:



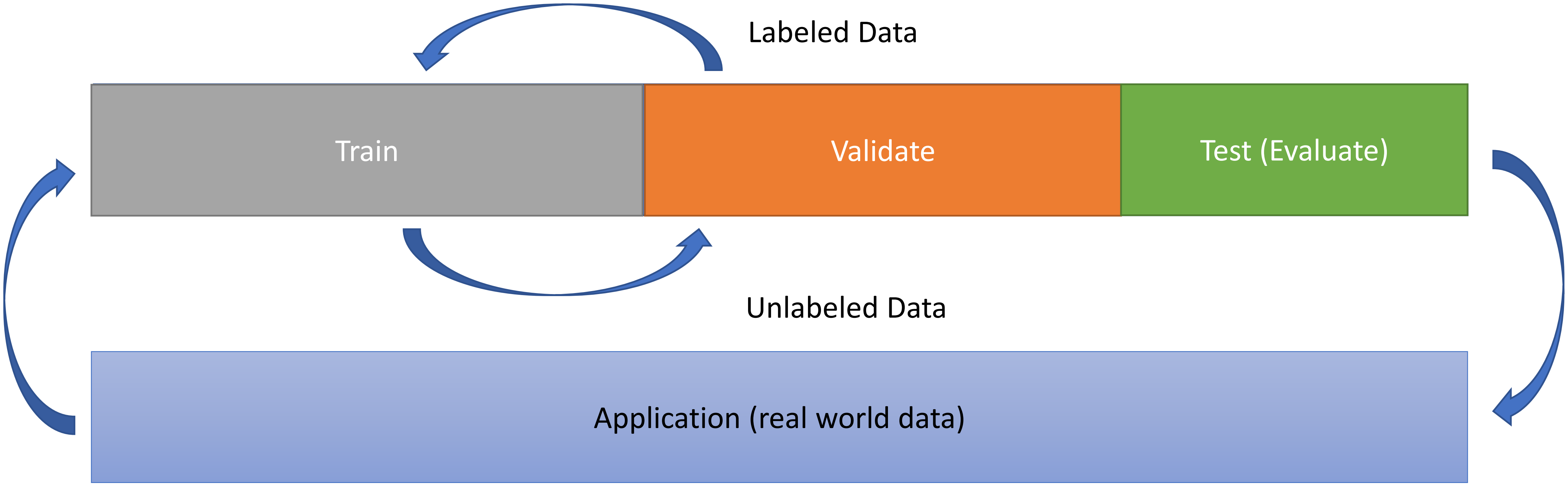
Revision Lecture

- Data partitioning:



Revision Lecture

- Data partitioning (sometimes, e.g. if more data available):



Revision Lecture

- Why is statistics essential in ML?
  1. Observations/Measurements can be statistically evaluated
  2. Predictions are always stochastically independent
  3. Knowledge can be statistically described
  4. Every ML algorithm is derived from Bayes' rule

Example

- Task: Hot Dog vs Not Hot Dog Image Classifier
- Used features: Presence of Logo
- Assumption: Data is representative

- What is the Probability of an image being a Hot Dog image based on this data, i.e. P(Hot Dog)?

No.	Logo?	Label
0	Yes	Hot Dog
1	Yes	Hot Dog
2	No	Hot Dog
3	Yes	Hot Dog
4	Yes	Not Hot Dog
5	Yes	Not Hot Dog
6	Yes	Hot Dog
7	Yes	Not Hot Dog
8	No	Not Hot Dog
9	Yes	Not Hot Dog

Example

- Task: Hot Dog vs Not Hot Dog Image Classifier
- Used features: Presence of Logo
- Assumption: Data is representative

- What is the Probability of an image not having a logo based on this data, i.e.
  P(No Logo)?

No.	Logo?	Label
0	Yes	Hot Dog
1	Yes	Hot Dog
2	No	Hot Dog
3	Yes	Hot Dog
4	Yes	Not Hot Dog
5	Yes	Not Hot Dog
6	Yes	Hot Dog
7	Yes	Not Hot Dog
8	No	Not Hot Dog
9	Yes	Not Hot Dog

Example

- Task: Hot Dog vs Not Hot Dog Image Classifier
- Used features: Presence of Logo
- Assumption: Data is representative

- What is the Probability of an image being a Hot Dog image based on this data if there is a logo on that image, i.e. P(Hot Dog | Logo)?

No.	Logo?	Label
0	Yes	Hot Dog
1	Yes	Hot Dog
2	No	Hot Dog
3	Yes	Hot Dog
4	Yes	Not Hot Dog
5	Yes	Not Hot Dog
6	Yes	Hot Dog
7	Yes	Not Hot Dog
8	No	Not Hot Dog
9	Yes	Not Hot Dog

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:

Sky Occlusion	Sky color	Class
Cloudy	Blue	Rainy
Cloudy	Red	Rainy
Cloudy	Blue	Rainy
Cloudy	Blue	Sunny
Clear	Blue	Sunny
Clear	Blue	Sunny

  - P(Sky Occlusion=Cloudy | Class = Sunny) = ?

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - P(Sky Occlusion=Clear | Class = Rainy) = ?

Sky Occlusion	Sky color	Class
Cloudy	Blue	Rainy
Cloudy	Red	Rainy
Cloudy	Blue	Rainy
Cloudy	Blue	Sunny
Clear	Blue	Sunny
Clear	Blue	Sunny

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - But what about continuous features?

Temp.	Class
19°C	Rainy
18°C	Rainy
20°C	Rainy
21°C	Sunny
22°C	Sunny
24°C	Sunny

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - But what about continuous features?

  - P(Temp=19.5°C | Class = Rainy) = ?

Temp.	Class
19°C	Rainy
18°C	Rainy
20°C	Rainy
21°C	Sunny
22°C	Sunny
24°C	Sunny

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - But what about continuous features?

  - P(Temp=19.5°C | Class = Rainy) = 0% ?

Temp.	Class
19°C	Rainy
18°C	Rainy
20°C	Rainy
21°C	Sunny
22°C	Sunny
24°C	Sunny

Revision Lecture

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - But what about continuous features?

  - P(Temp=19.5°C | Class = Rainy) = 0% ?

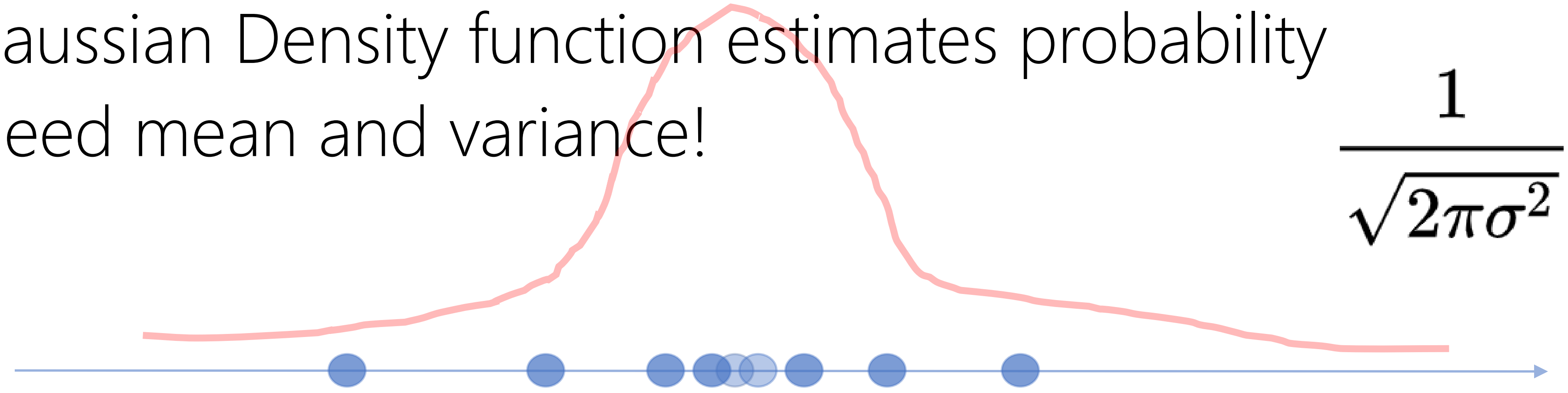
  - => Need to estimate underlying distribution!

Temp.	Class
19°C	Rainy
18°C	Rainy
20°C	Rainy
21°C	Sunny
22°C	Sunny
24°C	Sunny

Example

- We observe lots of continuous features and **assume** a gaussian distribution
- The most probable value should be where most data points are gathered, i.e. where it is densest.
- Gaussian Density function estimates probability
- Need mean and variance!

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Estimation of Likelihood

- In practice quite important:
  - Estimation of Likelihood P(feature | class)
  - Easy for categorical features:
  - But what about continuous features?

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  - P(Temp=19.5°C | Class = Rainy) = 0% ?

Temp.	Class
19°C	Rainy
18°C	Rainy
20°C	Rainy
21°C	Sunny
22°C	Sunny
24°C	Sunny

  - => Need to estimate underlying distribution!
  - Assume Gaussian
    => Mean Temp (Given Class = Rainy) = 19°C
    => Variance (Given Class = Rainy) = $\frac{2}{3}$

    => P(Temp=19.5°C | Class = Rainy) = $\frac{1}{\sqrt{2\pi\frac{2}{3}}} e^{-\frac{(19.5-19)^2}{2\cdot\frac{2}{3}}}$

Content

- Revision: McCulloch Pitts Neuron
- Revision Lecture
- Naive Bayes Classifier
- Tasks

Prerequisites

- Bayes' Formula

Likelihood

Prior

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)}$$

Posterior

Evidence

Prerequisites

- Chain Rule for joint probabilities:

$$P(c, x_1, x_2, \ldots, x_n) = P(c) \cdot P(x_1|c) \cdot P(x_2|c, x_1) \cdots P(x_n|c, x_1, \ldots, x_{n-1})$$

- Easy proof by induction

Naive Bayes Classifier

- Aim:

$$\underset{k=0,\ldots,N}{\operatorname{argmax}}[P(C_k|x_1,\ldots x_n)]$$

Naive Bayes Classifier

$$\underset{k=0,\dots,K}{\operatorname{argmax}}[P(C_k|x_1,\dots x_n)]$$

$$= \underset{k=0,\dots,K}{\operatorname{argmax}}[\frac{P(C_k)P(x_1,\dots x_n|C_k)}{P(x_1,\dots x_n)}]$$

$$= \underset{k=0,\dots,K}{\operatorname{argmax}}[\frac{P(C_k,x_1,\dots x_n)}{P(x_1,\dots x_n)}]$$

$$= \underset{k=0,\dots,K}{\operatorname{argmax}}[\{P(C_k,x_1,\dots x_n)]$$

Naive Bayes Classifier

$$= \underset{k=0,\ldots,K}{\operatorname{argmax}}[\{P(C_k, x_1, \ldots x_n)]$$

$$= \underset{k=0,\ldots,K}{\operatorname{argmax}}[P(C_k) \cdot P(x_1|C_k) \cdot P(x_2|C_k, x_1) \cdots P(x_n|C_k, x_1, \ldots, x_{n-1})]$$

Very naive assumption:

It is assumed, that **features** are stochastically independent from each other.
It means that knowing about the occurence of one feature, does not change the probability of any other feature, i.e. $P(x_i|x_j) = P(x_i)\ \forall i,\ i \neq j$

$$= \underset{k=0,\ldots,K}{\operatorname{argmax}}[P(C_k) \cdot P(x_1|C_k) \cdot P(x_2|C_k) \cdots P(x_n|C_k)]$$

Naive Bayes Classifier

- Given a labeled training set, how do we get these probabilities?

$$\underset{k=0,\ldots,K}{\operatorname{argmax}}[P(C_k) \cdot P(x_1|C_k) \cdot P(x_2|C_k) \cdots P(x_n|C_k)]$$

Prior of class C_k:
Number of class occurences in data set divided by number of all samples in data set

Likelihoods of all features, given class C_k

For each feature/class combination, we need a (gaussian) distribution model!
This way we can calculate the probability during inference!

Gaussian Naive Bayes Classifier

- „Training“:
  - Calculate class probabilities for all classes from training data
  - Calculate mean and standard deviation for each feature class combination
    (to model Gaussian each feature distribution given each class)

- Inference:
  - For each class calculate the product of likelihoods and the class prob
    (Use mean and standard deviation to estimate likelihood for input features)
  - Return class with highest value

$$\underset{k=0,\dots,K}{\operatorname{argmax}}[P(C_k) \cdot P(x_1|C_k) \cdot P(x_2|C_k) \cdots P(x_n|C_k)]$$

Content

- Revision: McCulloch Pitts Neuron
- Revision Lecture
- Naive Bayes Classifier
- Tasks