UNIVERSITÄT
ZU KÖLN

MACHINE LEARNING: INTRODUCTION

Sprachverarbeitung (Vorlesung)

Janis Pagel*

24 April 2025

*Based on slides by Nils Reiter

Introduction

- Collection of techniques for automatic
  - decision making
  - pattern detection
  - data analysis
- Machine learning vs. rule-based systems
  - Rule-based: Decision rules are hand-coded
    - if/then/else, ...
  - Machine learning: Decision ‘rules’ are ‘learned’ from data
  - Data is used to estimate weights and criteria

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
    if token[0].islower() and token.endswith("en"):
        return "VERB"
    elif token[0].isupper():
        return "NOUN"
    else:
        if token in determiners:
            return "DET"
...
```

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
    if token[0].islower() and token.endswith("en"):
        return "VERB"
    elif token[0].isupper():
        return "NOUN"
    else:
        if token in determiners:
            return "DET"
...
```

Which token properties are used here?

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
    if token[0].islower() and token.endswith("en"):
        return "VERB"
    elif token[0].isupper():
        return "NOUN"
    else:
        if token in determiners:
            return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
    if token[0].islower() and token.endswith("en"):
        return "VERB"
    elif token[0].isupper():
        return "NOUN"
    else:
        if token in determiners:
            return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

Which properties are *not* used?

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
    if token[0].islower() and token.endswith("en"):
        return "VERB"
    elif token[0].isupper():
        return "NOUN"
    else:
        if token in determiners:
            return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

Which properties are *not* used?

- Prefixes
- Token length
- Sequence: Previous tag

From Rules to Neural Networks

‘Classical’ machine learning

```
tokens = ["Der", "Hund", " bellt "]
tags = ["DET", "NOUN", "VERB"]

table = extract_features(tokens)

model = train(table, tags)
```

	Case	en-Suffix	In-Det-list
1	u	false	true
2	u	false	false
3	l	false	false

- Token properties → features
- Feature extraction / feature engineering
  - Finding useful features based on domain knowledge (e.g., linguistic knowledge)
  - ‘Playground’: What works well can really only be determined empirically

From Rules to Neural Networks

'Classical' machine learning

```
tokens = ["Der", "Hund", " bellt "]
tags = ["DET", "NOUN", "VERB"]

table = extract_features(tokens)

model = train(table, tags)
```

	Case	en-Suffix	In-Det-list
1	u	false	true
2	u	false	false
3	l	false	false

- Token properties → features
- Feature extraction / feature engineering
  - Finding useful features based on domain knowledge (e.g., linguistic knowledge)
  - 'Playground': What works well can really only be determined empirically
- Training: Estimate which features in which order allow best decisions
  - A large collection of algorithms has been developed: Decision trees, support vector machines, naive Bayes, ...
  - Training data needed: Words with manually assigned correct labels

From Rules to Neural Networks

Deep learning

- No more feature engineering
  - Models learn how to embed instances in vector space as their first step
- More compute cycles and more training data
- Black box
  - Intermediate states not interpretable for us humans
  - Only input and output can be understood

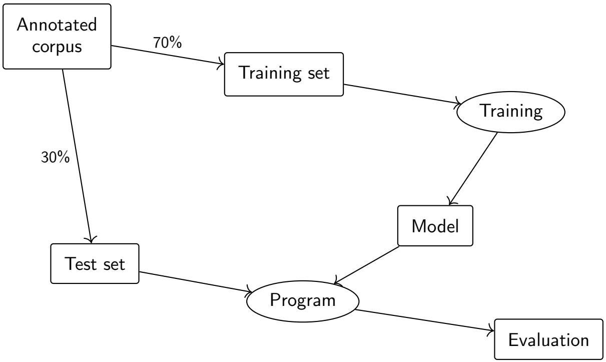
Development Stages

- Training
  - Estimate weights/features/rules based on annotated data
- Testing
  - Apply the model on annotated data
  - Estimate/calculate the correctness of its predictions
- Application
  - Train the model on as much data as possible
    - Assumption: More data ➔ Better results
  - Options: Evaluate in the wild, re-train based on usage data

Always separate train and test data

Training and Testing

- Goal: Apply the model on new data (and estimate its performance then)
- The program cannot have seen the data, so that it is a realistic test



Understanding Machine Learning

- Levels of understanding
  - Intuition
  - Formalization (math)
  - Implementation (code)
    - Complexity usually hidden in libraries

Understanding Machine Learning

- Levels of understanding
  - Intuition
  - Formalization (math)
  - Implementation (code)
    - Complexity usually hidden in libraries
- Areas to distinguish
  - Learning algorithm
  - Prediction model
  - Data preparation
    - Feature extraction (classical ML)
    - Shape of input data
  - Evaluation options

Classification

- Most straightforward task type
- Objects are categorized
- Categories (= classes) are known previously

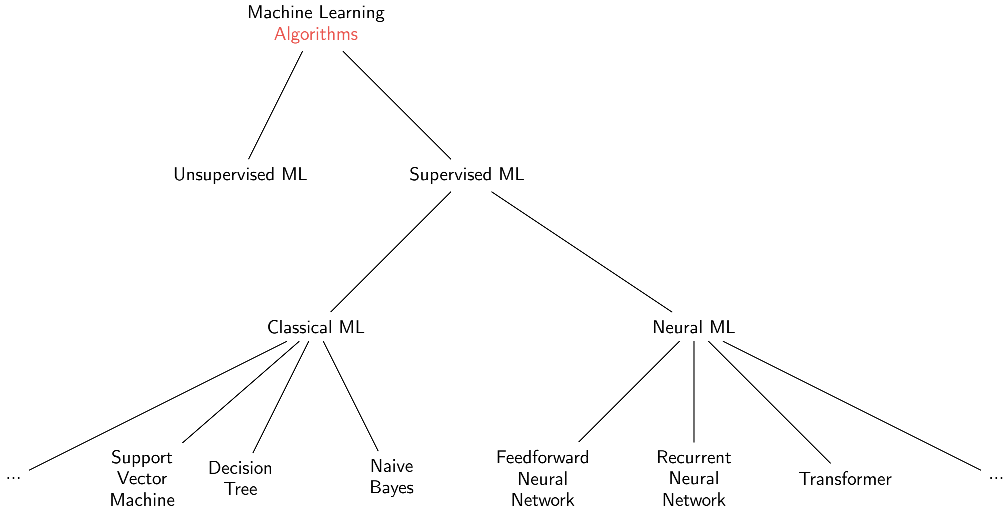
Classification

- Most straightforward task type
- Objects are categorized
- Categories (= classes) are known previously

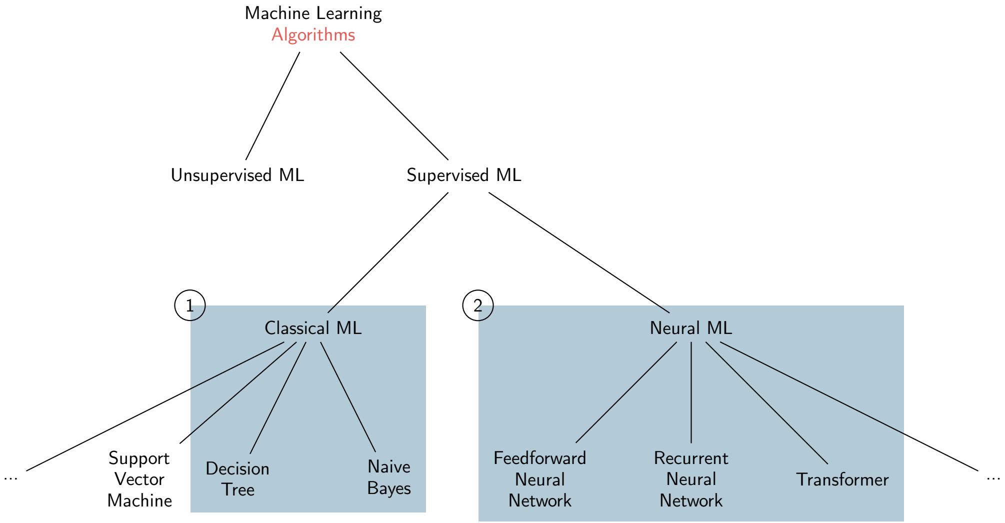
Examples

- Classify newspaper texts into genres (politics, economy, sports, ...)
- Classify reviews according to their opinion (positive, negative, neutral)
- Detect spam e-mail (classify mails in spam or not-spam)

Machine Learning



Machine Learning



Feature-Based Machine Learning

- How to represent our instances for the machine learning algorithm?
- Feature-based machine learning:
  - Humanly interpretable representations
  - Derived from knowledge about the domain in question
  - ML learns which properties of the data are relevant when and how
- These are called features

Features and Tasks

Examples

- Which features are relevant for detecting spam e-mail?
- Which features are relevant for detect plagiarism?
- Which features are relevant for assigning part of speech tags?

Features

- Used to describe classification items
- Feature extraction: Code to determine feature values for an item
- Features encode expected influence of item properties and target class
  - If we think a property could be relevant → make it a feature

Example

- Task: Assign part of speech information to words in context
  - "The dog barks." → (Det, Noun, Verb, Punct)
- Target class: Parts of speech (noun, verb, adjective, ...)

Features

- Used to describe classification items
- Feature extraction: Code to determine feature values for an item
- Features encode expected influence of item properties and target class
  - If we think a property could be relevant → make it a feature

Example

- Task: Assign part of speech information to words in context
  - “The dog barks.” → (Det, Noun, Verb, Punct)
- Target class: Parts of speech (noun, verb, adjective, ...)
- Features
  - Case (upper vs. lower)
  - Length
  - Suffix (last two characters)

Features

Data Types

Feature	Type
Case	
Length	
Suffix	

Features

Data Types

Feature	Type
Case	Three categories: upper/lower/other
Length	Integer
Suffix	String

Features

Feature Values

Word	Case	Length	Suffix	Class
The	upper	3	he	Det
dog	lower	3	og	Noun
barks	lower	5	ks	Verb
.	other	1	.	Punct

Table: Extracted features for example sentence, plus target class annotation

- This will be the input to the machine learning algorithm

Tables

- Tables are the backbone of quantitative analysis
- Convention: Items in rows, properties/features in columns

Tables

- Tables are the backbone of quantitative analysis
- Convention: Items in rows, properties/features in columns
- Main data types: Numbers, categories
  - If all entries are numeric, it's a (mathematical) matrix
- Various file formats
  - CSV/TSV: Comma/tab-separated values
  - XLS/XLSX: Excel format
    - Because the file format is proprietary, not used for exchange or archival

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower ,3, og,Noun
barks , lower ,5, ks , Verb
 ., other  ,1,., Punct
```

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower,3,og,Noun
barks,lower,5,ks,Verb
.,other,1,.,Punct
```

- Plain text files
- Items separated by newline, feature values by comma
- Problems?

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower,3,og,Noun
barks,lower,5,ks,Verb
.,other,1,.,Punct
```

- Plain text files
- Items separated by newline, feature values by comma
- Problems? What if the sentence contains a comma?

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower,3,og,Noun
barks,lower,5,ks,Verb
.,other,1,.,Punct
```

- Plain text files
- Items separated by newline, feature values by comma
- Problems? What if the sentence contains a comma?
  - Escaping: Use special characters without their special meaning: \,

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower,3,og,Noun
barks,lower,5,ks,Verb
.,other,1,.,Punct
```

- Plain text files
- Items separated by newline, feature values by comma
- Problems? What if the sentence contains a comma?
  - Escaping: Use special characters without their special meaning: \,
  - Quoting: Enclose them in quote characters ","

Comma-Separated Values (CSV)

```
The,upper,3,he,Det
dog,lower,3,og,Noun
barks,lower,5,ks,Verb
.,other,1,.,Punct
```

- Plain text files
- Items separated by newline, feature values by comma
- Problems? What if the sentence contains a comma?
  - Escaping: Use special characters without their special meaning: \,
  - Quoting: Enclose them in quote characters ","
- Different strategies, all are used

UNIVERSITÄT ZU KÖLN

Tab-Separated Values (TSV)

Code Listing 1: A TSV representation, with tabs represented as →

```
The→upper→3→he→Det
dog→lower→3→og→Noun
barks→lower→5→ks→Verb
.→other→1→.→Punct
```

- Similar to CSV, but with a tab instead of a comma
- Tab character: A single character with variable width
  - Often used for indentation
- Escaped with \t (e.g., in regular expressions)

Tab-Separated Values (TSV)

Code Listing 2: A TSV representation, with tabs represented as →

```
The→upper→3→he→Det
dog→lower→3→og→Noun
barks→lower→5→ks→Verb
.→other→1→.→Punct
```

- Similar to CSV, but with a tab instead of a comma
- Tab character: A single character with variable width
  - Often used for indentation
- Escaped with \t (e.g., in regular expressions)
- CSV/TSV have undefined 'edge cases'
  - Escaping, quoting, comments
  - Inspect before processing

CSV/TSV Tools

- Most spreadsheets programs can import and export CSV/TSV (MS Excel, Apple Numbers, Google Spreadsheets, OpenOffice Calc)

CSV/TSV Tools

- Most spreadsheets programs can import and export CSV/TSV (MS Excel, Apple Numbers, Google Spreadsheets, OpenOffice Calc)

Reading/writing CSV

- Java: Apache Commons CSV `https://commons.apache.org/proper/commons-csv/`
- Python: Module in standard library `https://docs.python.org/3/library/csv.html`
- Command line
  - csvkit `https://csvkit.readthedocs.io/en/latest/`
  - awk/gawk `https://www.gnu.org/software/gawk/manual/gawk.html`

XLS/XLSX

- File format used by MS Excel
- Binary, closed
- Don't use Excel as a database: `https://www.youtube.com/watch?v=zUp8pkoeMss`

XLS/XLSX

- File format used by MS Excel
- Binary, closed
- Don't use Excel as a database: `https://www.youtube.com/watch?v=zUp8pkoeMss`
- Useful for lightweight calculation/visualisation
- Difficult to integrate with other tools

CoNLL-Format

- Often used in natural language processing
- Similar to CSV with one token per line, but
  - Row order shows token order
  - Empty lines indicate sentence boundaries
  - What is exactly in each column differs: CoNLL != CoNLL
    - `https://universaldependencies.org/format.html`
    - `https://cemantix.org/conll/2012/data.html`

Data Types

CSV/TSV files

- Everything is a string
- If you import/read a CSV table, you need to convert things into appropriate data types
- Potential error source:
  If you inspect the beginning of a long table and find that column 5 contains integer values – it could still be the case that at some point column 5 contains something else.
  There are no guarantees!

Preparation Steps

Data Analysis

- Important to get to know your data set
  - How many instances are there?
  - How are the classes distributed?
  - Text features: How long are they (min/max/average)? Are they categories or free text?
  - Numeric features: What's their distribution? Does the enumeration encode something?

Preparation Steps

Data Analysis

- Important to get to know your data set
  - How many instances are there?
  - How are the classes distributed?
  - Text features: How long are they (min/max/average)? Are they categories or free text?
  - Numeric features: What's their distribution? Does the enumeration encode something?

Preprocessing

- Light-weight processing before training and during development
- Typical tasks: Casing, stop words, lemmatization

Summary

- Machine learning: Let the machine figure out which properties are relevant when
- Feature-based ML: Humans define domain-specific features
- Neural ML: Machine *also* figures out which features to use
- Train and test data
- ML data often comes in tables
- Preparatory steps: Data analysis and preprocessing
- Next session: How to evaluate ML systems