# Recap

- Last Tuesday: Running long processes
  - Executing `exit` terminates all running processes
  - We often want to have something run for some time
  - `tmux` allows us to have a persistent session on a server
- Last Thursday: Classical language models ($n$-gram language models)
  - Predict the next word given some history – $p(word|history)$
  - Limited history
  - Probabilities estimated on corpus, using maximum likelihood estimation (MLE)
  - Smoothing to prevent zero probabilities

# ML: Data Sets and File Formats

## Sprachverarbeitung (VL + Ü)

Nils Reiter

May 9, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

## Today: Classification

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

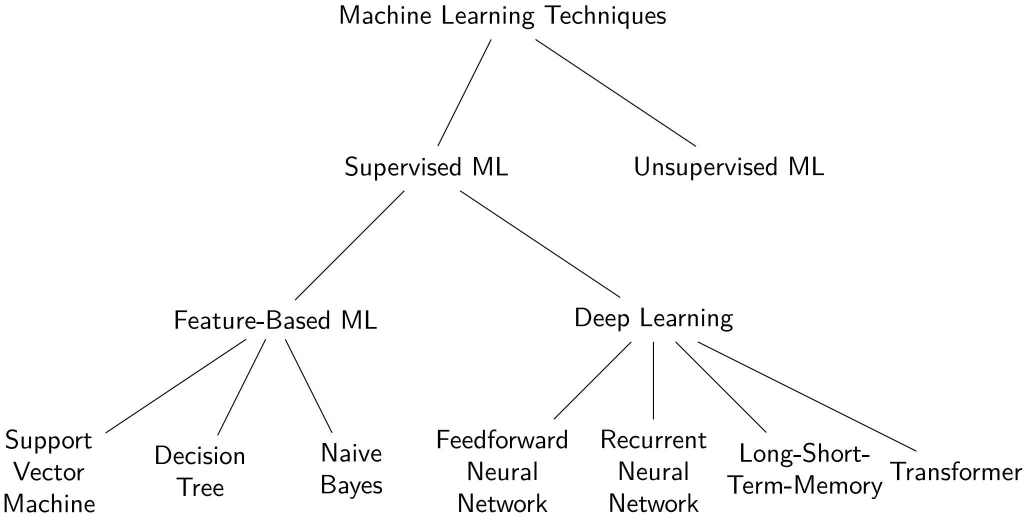
# Today: Classification

- 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 Techniques

Supervised ML

Unsupervised ML

Feature-Based ML

Deep Learning

Support Vector Machine

Decision Tree

Naive Bayes

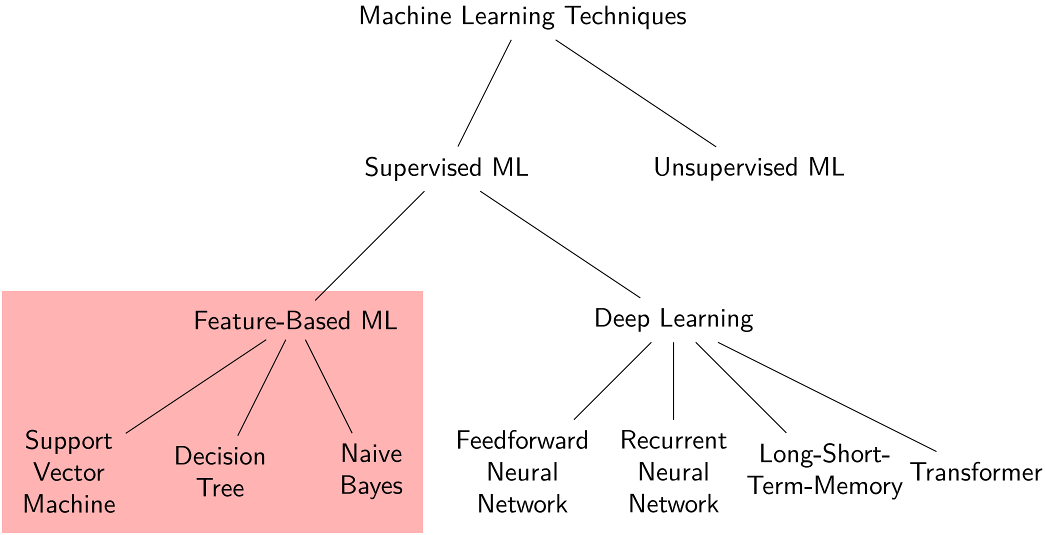
Feedforward Neural Network

Recurrent Neural Network

Long-Short-Term-Memory

Transformer

# Machine Learning



# Feature-Based Machine Learning

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

# Feature-Based Machine Learning

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

### Example

- Which properties are relevant for detecting spam?

# Feature-Based Machine Learning

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

## Example

- Which properties are relevant for detecting spam?
  - Certain keywords in the text: »casino«, »enlargement«, »drugs«, …
  - Metadata
    - Is the sender known (= in our address book)?
    - Is the server known to be used by spammers?
    - Are we in the `To` field (or blind copied)?

# 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
    - ARFF: Weka file format (= CSV + type declarations)

# 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

# Tab-Separated Values (TSV)

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
- Can be escaped with `\t` (e.g., in regular expressions)

# Tab-Separated Values (TSV)

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
- Can be 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

## ARFF

- Used by machine learning toolkit Weka
- Data as CSV
- Header to define attributes/features
- Name/type for each attribute
  - Nominal values: Possible values

# 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!

# 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!

ARFF

- Data types are declared in the header

# Section 2

## Exercise

# Exercise

The file /teaching/summer-2023/sprachverarbeitung/data/titanic.csv contains passenger data from the crossing of the Titanic, which sank in the North Atlantic on 15 April 1912.[1] The file contains information on the passengers, together with details of who survived the sinking and who did not.

The following tasks require a mixture between reading/thinking and applying command line tools. All tasks can be solved purely manual, but this is cumbersome and takes a lot of time.

1. Find out what features are given for the passengers.
2. Assign a data type to each feature.
3. For category features: Find out which categories there are, and how common each is.
4. For numeric features: Find out what range of values they assume (minimum and maximum).

**Hints**

- One column is particularly tricky, because it contains relatively free text, including commas and quotation marks. Maybe first clean up this column by removing commas etc. or remove the column entirely. Both can be done with regular expressions as we have used them before.
- Accessing individual columns could be done with a similar trick as when we defined the context for the concordance in words instead of characters: Define some content of the column and the character that separates the columns as a group, and ask this group to be repeated.

[1] https://en.wikipedia.org/wiki/Titanic