Introduction

Jan Sedmidubský Masaryk University

sedmidubsky@mail.muni.cz

Outline

- Course information
  - Objectives of the course
  - Evaluation – semestral project + final exam
  - Outline of the lectures + what I can learn in other courses
- Data mining pipeline
  - Data preprocessing
  - Tasks – classification, regression, prediction, event detection, anomaly detection
  - Learning – supervised, self-supervised, semi-supervised, unsupervised, active, meta
- Semestral project in detail – conditions and tasks
- Existing machine-learning tools/libraries
  - Deep learning frameworks – TensorFlow, Keras, PyTorch

Course objectives

- Learn principles of selected machine-learning (ML) and data-mining (DM) techniques
- Understand how selected techniques can be applied to specific real-life use cases
- Solve practical tasks within a group of students (semestral projects)

Course evaluation

- Final exam: 80%
  - Open questions, focus on main principles + applicability
- Semestral project: 20%
  - 3–4 students for one group
  - Goal – solve a given machine-learning/data-mining problem
    - E.g., classification of plant-disease images
  - You are expected to:
    - Implement your solution using the Google Colab environment (cloud Jupyter Notebooks)
    - Write a 2-page project report
    - Present your project (10 minutes presentation + 5 minutes discussion)
  - Details specified later
    - Project organizer: Ondřej Sotolář (xsotolar@fi.muni.cz)

Course topics

1) Introduction to machine learning and data mining + projects
2) Metric learning, product quantization, approximate searching
3) Advanced clustering methods
4) Advanced anomaly detection
5) Bayesian optimization
6) Automated machine learning
7) Time-series data mining
8) Processing of multidimensional time series of human motion
9) Cross-modal learning
10) Applied machine learning: examples of real-life applications

LLMs as a personal tutor

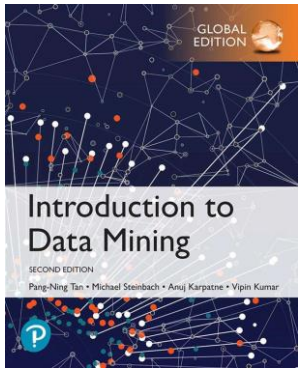
- You can use LLMs (e.g., ChatGPT) to
  - Discuss suitable methods and parameter settings for different use cases
  - Generate and debug Python code for experimenting with the methods
  - Generate multiple-choice and open questions for self-assessment

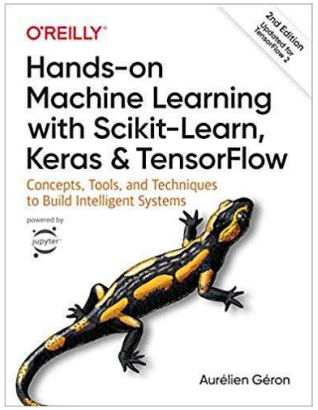
Source: New York Times

OpenAI

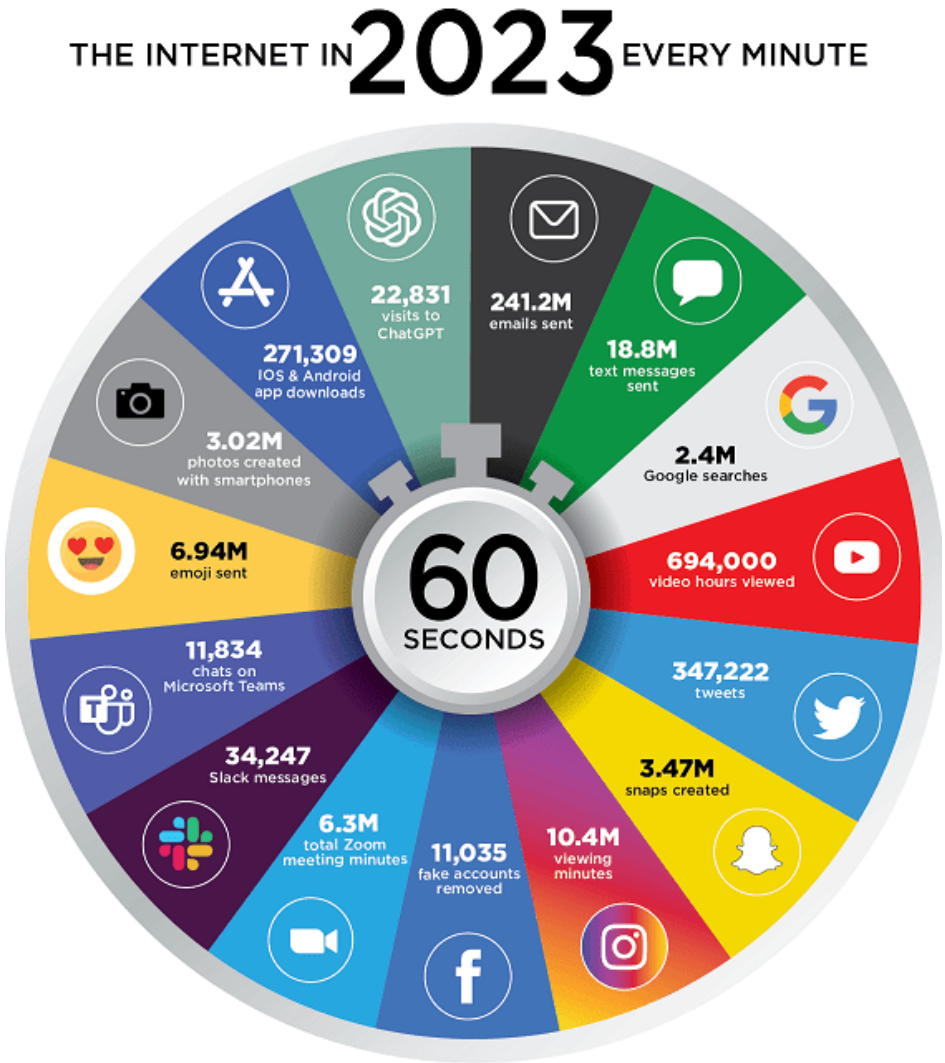
Literature and sources

- Textbooks:
  - Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining. 2nd Edition. Pearson / Addison Wesley, 2019.
  - Aurélien Géron: Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. 2nd or 3rd Edition, O'Reilly, 2019 or 2022
- Other sources:
  - University of Mannheim
    - Introduction to Data Mining by Christian Bizer
  - University of Minnesota
    - Introduction to Data Mining by Tan, Steinbach, Karpatne, Kumar
  - Purdue University
    - Deep Learning by Avi Kak and Charles Bouman





Big data everywhere



Big data everywhere

- US Library of Congress: ≈ 235 TB archived ≈ 40 Wikipedias
- arXiv Preprint Server: > 2 million papers
- Tasks:
  - Discover topic distributions or citation networks
  - Train Large Language Models

- Facebook
  - 4 Petabyte of new data generated every day
  - over 300 Petabyte in Facebook's data warehouse
- Tasks:
  - Predict interests and behavior of over one billion people

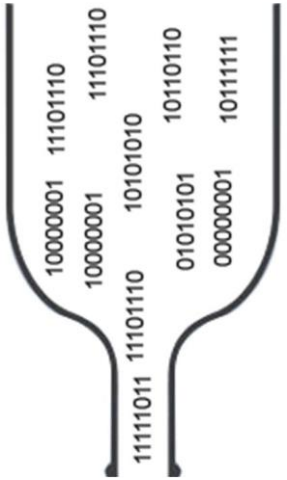
Big data everywhere

- Law enforcement agencies
  - Collect unknown amounts of data from various sources
    - Cell phone calls
    - Location data
    - Web browsing behavior
    - Credit card transactions
    - Online profiles (Facebook)
    - …
- Tasks:
  - Predict terrorist
  - Find compromising photos

Source: https://www.novinky.cz/clanek/krimi-fbi-odhalila-mladeho-cecha-ktery-vyhrozoval-odpalenim-bomby-40391731

Data mining

- Data mining – process of discovering patterns, relationships, and insights from large datasets
  - Goal – extract useful information from raw data
- DM methods help us to take decisions based on the patterns



← Amount of data that is collected

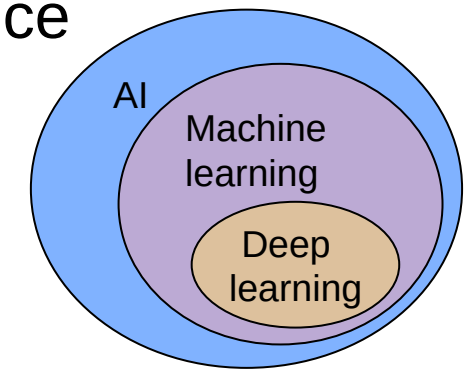
← Amount of data that can be looked at by humans

Exploration & analysis of large quantities of data in order to discover meaningful patterns

Non-trivial extraction of implicit, previously unknown, and potentially useful information from data

Machine learning

- Machine learning – branch of AI that enables computers to learn from data and make predictions/decisions without explicit programming
  - Goal – develop models that improve performance automatically through experience



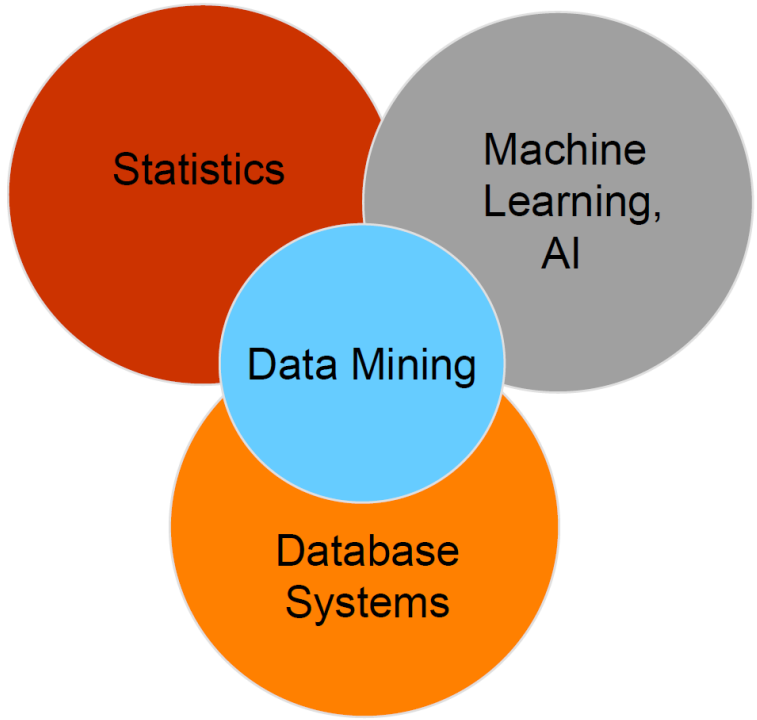
Statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions

Improving performance on a specific task by recognizing patterns, making predictions or decisions based on input data

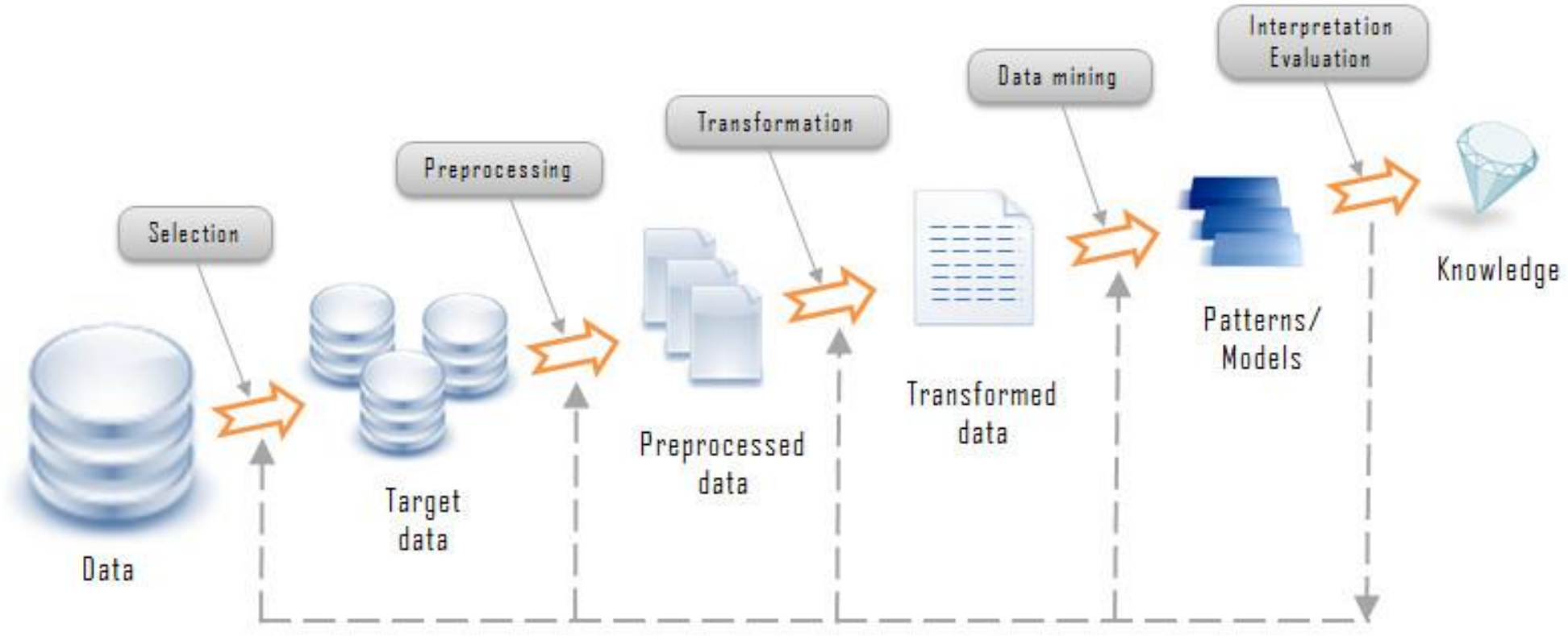
- Key components:
  - Experience – input data or historical information the system learns from
  - Task – the specific problem the system is trying to solve (e.g., image classification, speech recognition)
  - Performance measure – a metric used to evaluate how well the system performs the task (e.g., accuracy, precision, recall)

Data mining vs. machine learning

	Data mining	Machine learning
Purpose	Find patterns & insights	Make predictions & automate decisions
Approach	Exploratory analysis	Algorithm-based learning
Dependence on humans	More human-driven (analysis & interpretation)	More automated (self-improving models)
Outcome	Knowledge discovery	Predictive modeling
Example	Market basket analysis (which products are bought together)	Recommender system (suggesting products based on user behavior)



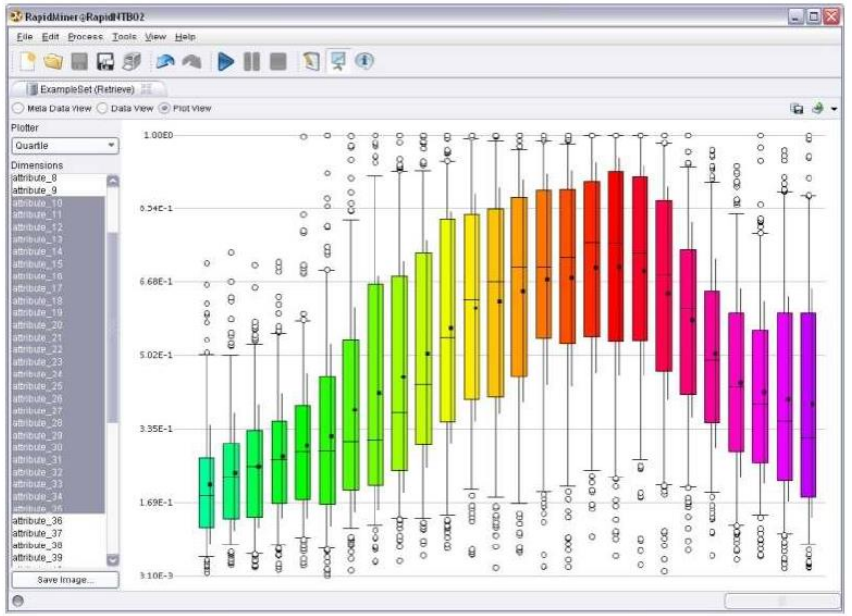
Data mining pipeline

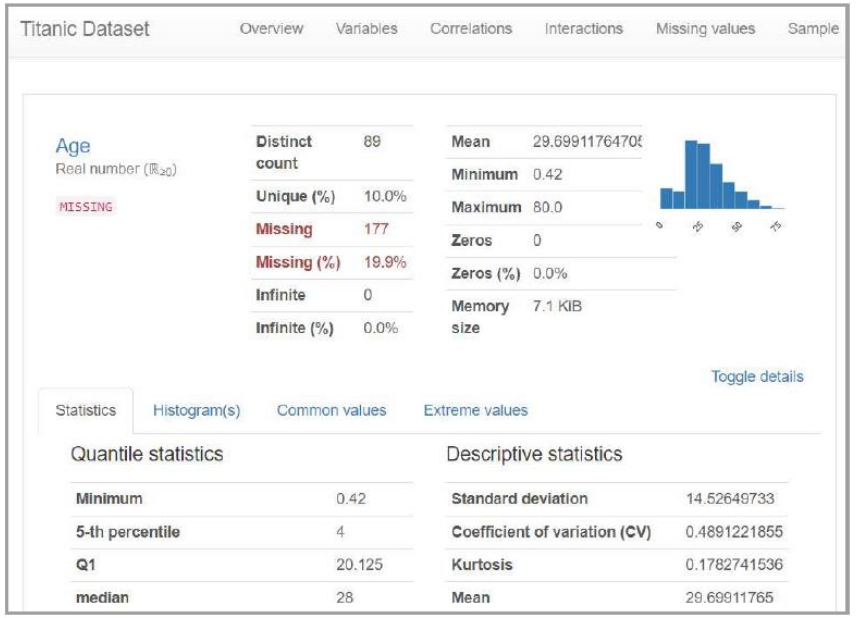


Source: https://www.linkedin.com/pulse/data-mining-knowledge-discovery-process-model-leandro-guerra/

1) Data selection

- Data types – text, images, videos, audio, time-series, spatio-temporal data, etc.
- Selection
  - What data is potentially useful for the task at hand?
  - What data is available?
  - What do I know about the quality of the data?
- Exploration / profiling
  - Get an initial understanding of the data
  - Calculate basic summarization statistics
  - Visualize the data
  - Identify data problems such as outliers, missing values, duplicate records





2) Preprocessing and transformation

- Data cleaning – handling missing values, removing duplicate records
- Transformation of data into a suitable representation
  - Scales of attributes (nominal, ordinal, numeric)
  - Number of dimensions (represent relevant information using less attributes)
  - Amount of data (determines hardware requirements)
- Methods
  - Discretization and binarization
  - Feature subset selection / dimensionality reduction
  - Attribute transformation / text to term vector / embeddings
  - Aggregation, sampling
  - Integration of data from multiple sources
- Good data preparation is key to producing valid and reliable models
  - Data integration/preparation takes 70–80% of the time and effort

3) Data mining

- Input: preprocessed data
- Output: model / patterns
- Steps:
  1) Apply data mining method
  2) Evaluate resulting model / patterns
  3) Iterate
     - Experiment with different hyperparameter settings
     - Experiment with multiple alternative methods
     - Improve preprocessing and feature generation
     - Increase amount or quality of training data
  4) Deploy – use the most promising model in the business context

Tasks and applications

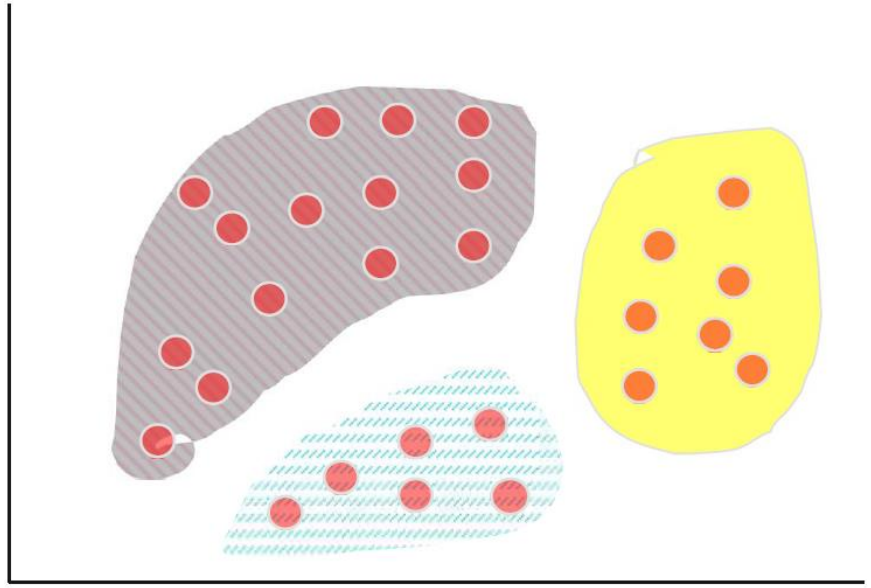
- Descriptive tasks
  - Goal: Find human-interpretable patterns that describe the data
  - Example: *Which products are often bought together*?
- Predictive tasks
  - Goal: Use some variables to predict unknown or future values of other variables
    - Given observations (e.g., from the past)
  - Example: *Will a person click an online advertisement*?
    - Given their browsing history
- Machine learning terminology
  - Descriptive ~ unsupervised
  - Predictive ~ supervised

Tasks

- Cluster analysis [Descriptive]
- Classification [Predictive]
- Regression [Predictive]
- Association analysis [Descriptive]
- Anomaly detection [Predictive]
- Time-series forecasting [Predictive]
- Event detection [Predictive]
- (Cross-modal) retrieval [Descriptive]

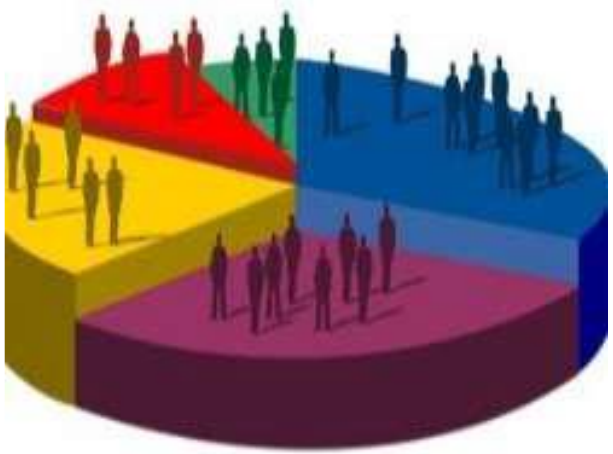
Cluster analysis

- Goal: given a set of data points, each having a set of attributes, and a similarity measure among them, find groups such that
  - Data points in one group are more similar to one another
    - Intra-cluster distances are minimized
  - Data points in separate groups are less similar to one another
    - Inter-cluster distances are maximized
- Similarity measures
  - Euclidean distance if attributes are continuous
  - Other task-specific similarity measures
- Result: a descriptive grouping of data points



Cluster analysis – example

- Application area: market segmentation
- Goal: find groups of similar customers
  - Group may be conceived as a marketing target to be reached with a distinct marketing mix
- Approach:
  1) Collect information about customers
  2) Find clusters of similar customers
  3) Measure the clustering quality by observing buying patterns after targeting customers with distinct marketing mixes



Classification

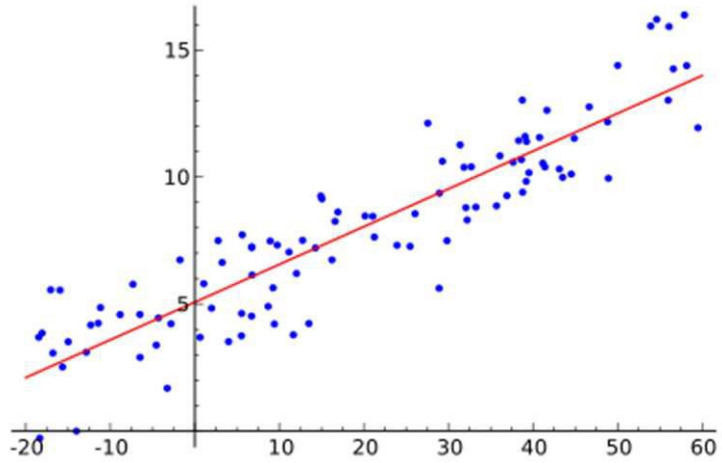
- Goal: previously unseen records should be assigned a class from a given set of classes as accurately as possible
- Approach:
  1) Given a collection of records (training set)
     - Each record contains a set of attributes
     - One attribute is the class attribute (label) that should be predicted
  2) Find a model for predicting the class attribute as a function of the values of other attributes

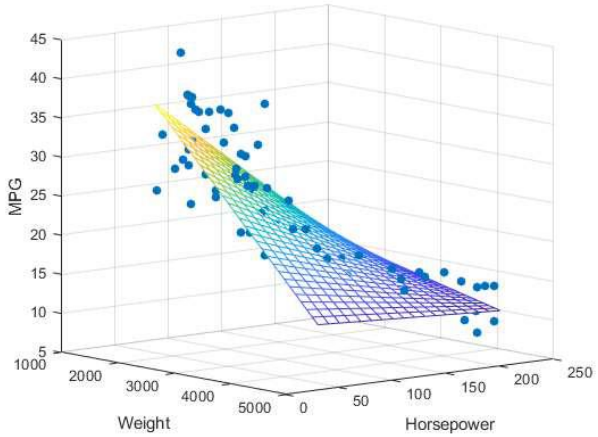
Classification – example

- Application area: fraud detection
- Goal: predict fraudulent cases in credit card transactions
- Approach:
  1) Use credit card transactions and information about account-holders as attributes
     - When and where does a customer buy? What does he buy?
     - How often he pays on time? Etc.
  2) Label past transactions as fraud or fair transactions
     - This forms the class attribute
  3) Learn a model for the class attribute from the transactions
  4) Use this model to detect fraud by observing credit card transactions on an account

Regression

- Goal: predict a value of a continuous variable based on the values of other variables, assuming a linear or nonlinear model of dependency





- Examples – predicting:
  - The price of a house or car
  - Sales amounts of new product based on advertising expenditure
  - Miles per gallon (MPG) of a car as a function of its weight and horsepower
  - Wind velocities as a function of temperature, humidity, air pressure, etc.
- Difference to classification: the predicted attribute is continuous, while classification is used to predict nominal attributes (e.g., yes/no)

Association analysis

- Goal: given a set of records each of which contain some number of items from a given collection, discover frequent itemsets
  - Produce association rules which will predict occurrence of an item based on occurrences of other items

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Frequent Itemsets
{Diaper, Milk, Beer}
{Milk, Coke}

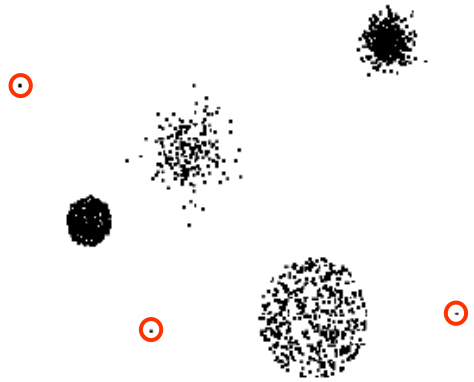
Association Rules
{Diaper, Milk} --> {Beer}
{Milk} --> {Coke}

Association analysis – example

- Application area: supermarket shelf management
- Goal: identify items that are bought together by sufficiently many customers
- Approach: process the point-of-sale data collected with barcode scanners to find dependencies among items
- A classic rule and its implications:
  - If a customer buys diapers and milk, then they are likely to buy beer as well
    - So, don’t be surprised if you find six-packs stacked next to diapers
    - Promote diapers to boost beer sales
    - If selling diapers is discontinued, this will affect beer sales as well

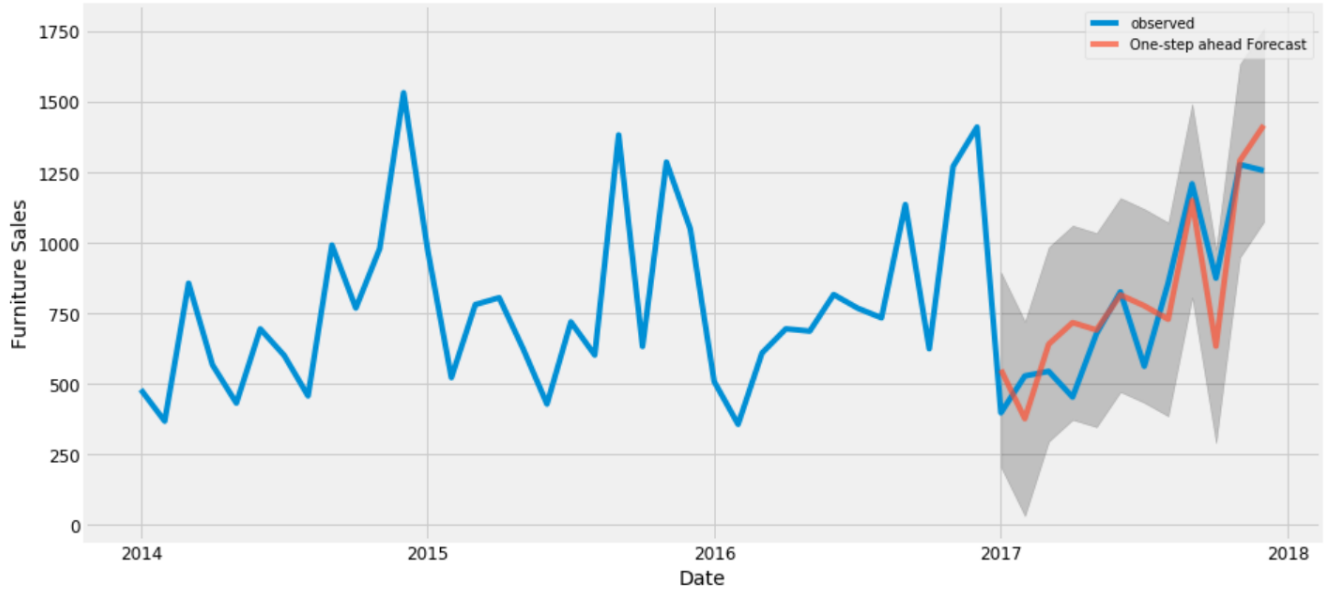
Deviation / anomaly detection

- Goal: detect significant deviations from normal behavior
- Examples:
  - Network intrusion detection
  - Identify anomalous behavior from sensor networks for monitoring and surveillance
  - Detecting changes in the global forest cover



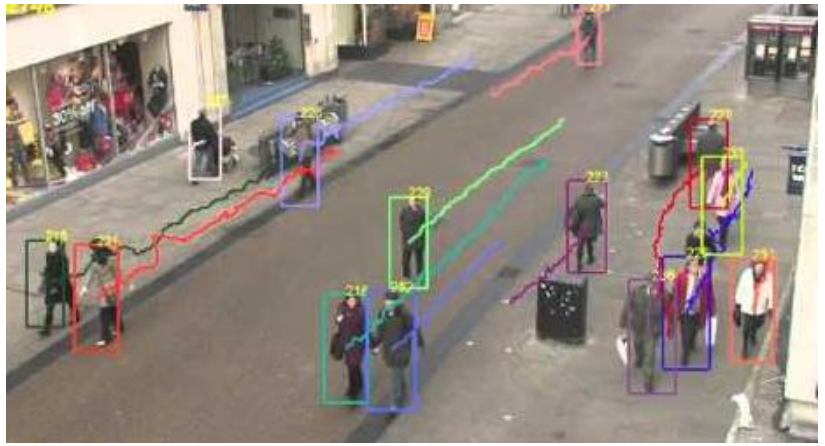
Time-series forecasting

- Goal: predict future values of a series of data points based on historical data, which is typically organized chronologically
- Examples:
  - Predict future demand for products to optimize inventory and reduce costs
  - Predict energy usage to balance supply and demand effectively
  - Forecast stock prices, currency exchange rates, or economic indicators



Event detection

- Goal: identify, classify, and analyze significant occurrences or patterns in data streams
  - These events often represent meaningful changes, anomalies, or predefined patterns in the data
- Examples:
  - Monitor network activity to identify suspicious or unauthorized access attempts
  - Identify trending topics, significant news in real-time from social media platforms
  - Detect suspicious human-motion actions (e.g., kicking) from surveillance cams



Cross-modal retrieval

- Goal: retrieve relevant data from one modality (e.g., text, image, audio, or video) using a query from another modality
  - This enables seamless interaction between different types of data, leveraging the relationship between them to deliver meaningful results
- Examples:
  - Retrieve unannotated images based on textual descriptions or vice versa
  - Index large video datasets (e.g., YouTube, Netflix) for content-based retrieval
  - Retrieving clinical notes based on visual annotations or imaging results

Learning

- Supervised
- Semi-supervised
- Unsupervised (self-supervised)
- Active learning
- Meta learning

Learning

- Supervised learning
  - Learning from a labeled dataset where the input-output relationship is known
  - Key features:
    - Data has labels
    - Model learns a mapping function (e.g., classification or regression tasks)
  - Examples: image classification, speech recognition
  - Challenges: requires a large amount of labeled data

Learning

- Unsupervised (self-supervised) learning
  - Learns patterns from unlabeled data
  - Key features:
    - No labeled data
    - Focuses on clustering, dimensionality reduction, and anomaly detection
  - Examples: clustering customers into segments, discovering hidden patterns

Learning

- Semi-supervised learning
  - Combines a small amount of labeled data with a large amount of unlabeled data
  - Key features:
    - Uses both labeled and unlabeled data
    - Improves performance when labeled data is scarce
  - Examples: text classification where only a few labeled examples are available, but a large amount of raw text can be leveraged

Learning

- Active learning
  - Model actively queries for labels in the data it is most uncertain about
  - Key features:
    - Reduces labeling costs by asking for human annotations only on difficult or ambiguous samples
  - Examples: real-world scenarios where labeling all data is expensive, such as medical diagnosis

Learning

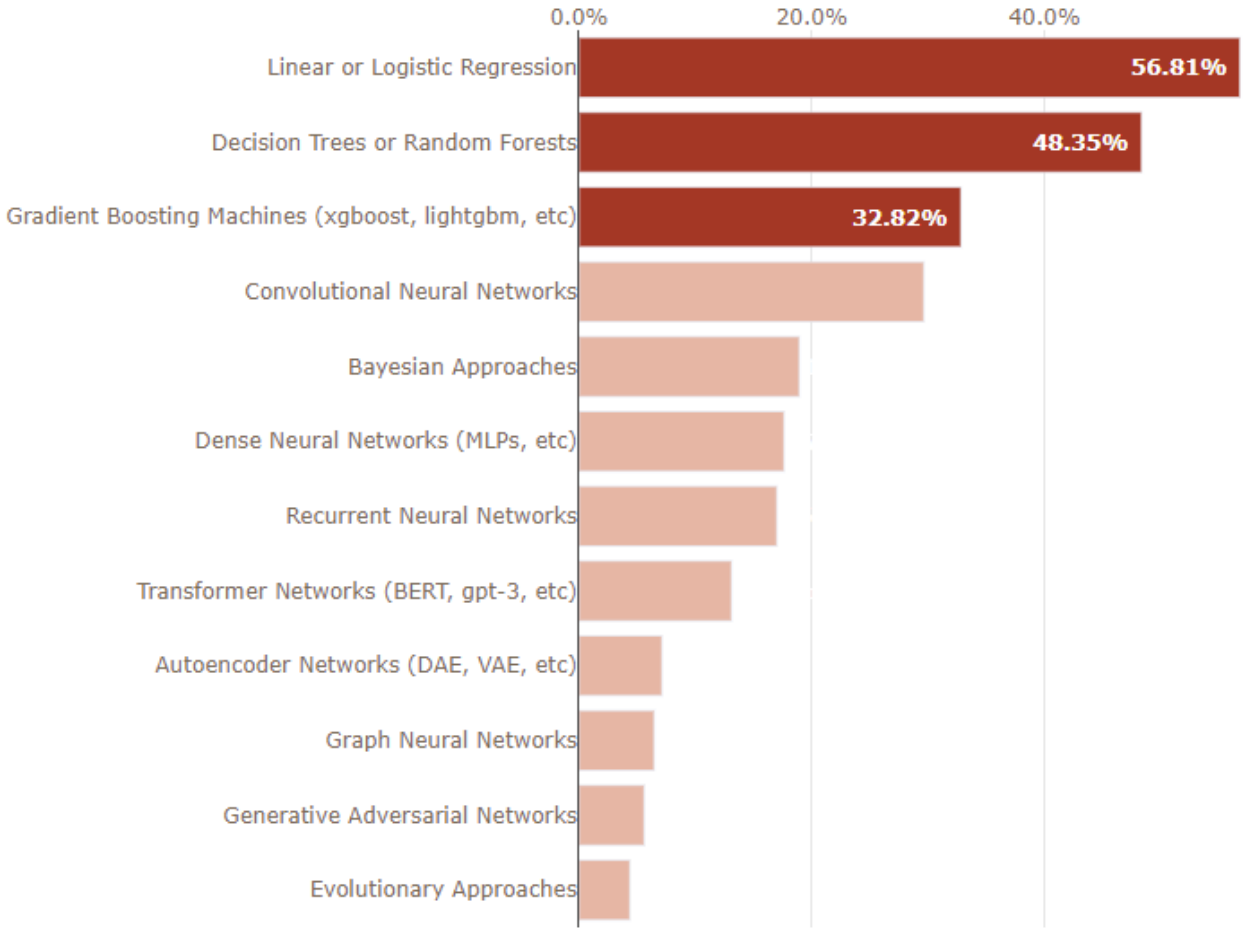
- Meta learning
  - Learning to learn – the model adapts to new tasks by leveraging past learning experiences
  - Key features:
    - Focuses on fast learning from few examples
    - Helps in generalizing to new tasks quickly
  - Examples: few-shot learning where the model learns to classify from only a few examples per class
    - A model trained on various handwriting styles can quickly adapt to recognizing a new, unseen script with minimal samples

Learning – comparison

Learning type	Data labels	Purpose	Example applications
Supervised learning	Labeled	Maps input to output	Image classification
Semi-supervised	Mixed	Leverages limited labeled data	Text categorization
Unsupervised	Unlabeled	Finds hidden structures or patterns	Clustering, dimensionality reduction
Active Learning	Minimal	Queries only the most uncertain data points	Medical image labeling
Meta Learning	Few labeled	Learns how to learn new tasks faster	Few-shot classification

Top ML algorithms in industry

- The reasons for use:
  - High accuracy for structured data
  - Easy to implement and train
  - Interpretability and explainability



Kaggle online poll 2022, 23,997 respondents,

Source: https://www.kaggle.com/code/eraikako/data-science-and-mlops-landscape-in-industry

Semestral project

- Goal – design, implement and test some ML/DM task
- Requirements – you will:
  - Select one of the offered topics
  - Form a team of 3–4 students and collaborate
  - Implement your solution in **Google Colab**
    - You provide a link to your solution in Colab
  - Write a compact technical report with hard limit of 2 pages
    - You upload the report into IS MU
  - Present your project within a 10-minute presentation
- Limitations:
  - If you decide to use prompt engineering, you need to include at least 2 additional techniques in your solution, such as RAG or prompt augmentation through external signals

Semestral project – textual report

- Write a compact technical report with hard limit of 2 pages + appendices (additional plots or tables, author contributions)
  - Use the Springer template – LaTeX or Word
  - It should contain the following sections:
    - Introduction
    - Related work
    - Proposed method(s)
    - Results and discussion
    - Appendix
      - Author Contributions: very short descriptions of individual author contributions
  - Recommendation – the report should include one table/plot with results, additional tables/plots can be included in Appendix or in the Colab notebook

Semestral project – deadlines

- Feb 18–28: forming groups of 3–4 students + topic selection
  - Task: enter information to provided Excel Sheet
- March 1–April 15: implementation phase (i.e., deadline **April 15**)
  - If you have any issue, you can ask for feedback (Ondřej Sotolář: xsotolar@fi.muni.cz)
  - Task: handover the link to your Colab solution & report PDF into IS MU vault
    - Test if the notebook is set to shared
- April 15–29: preparation of presentation
  - Task: prepare 10-minute presentation – presentations starting from **April 29**
    - Shortly: introduce your problem & related work, mainly focus on your approach and results
- Evaluation – you will be notified about the final score which constitutes 20% of the final mark
  - 14% for a basic solution, 6% bonus for addressing the reviewer's issues or high-quality work

Semestral project – topics

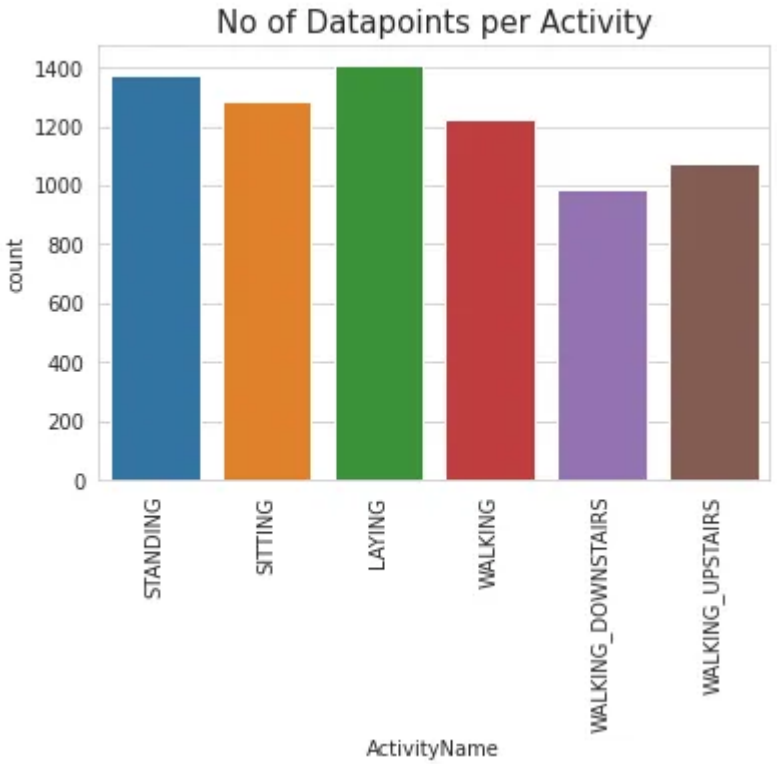
- Topic1 – Human activity recognition (~time series classification)
  - https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones
- Topic2 – Food hazard detection (~multi-modal text classification)
  - Optionally muti-modal
  - https://food-hazard-detection-semeval-2025.github.io
- Topic3 – Plant disease classification (~image classification)
  - https://github.com/Denisganga/the_plant_doctor/tree/main
- Topic4 (harder) – Urban sound classification (~signal processing)
  - https://www.kaggle.com/code/aadith0/rnn-audio-classification

Topic1: Human Activity Recognition (HAR)

- Classify recordings of study participants performing activities while carrying a smartphone with embedded inertial sensors
- Data:
  - https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones
- Data analysis:
  - https://www.kaggle.com/code/anushareddy56/starter-human-activity-recognition-6cad9ae9-2
  - https://sakshamchecker.medium.com/human-activity-recognition-7abaa9a1cf34
- Baseline: naïve LSTM ~0.8 $F1_{macro}$
  - https://colab.research.google.com/drive/1a1QP9gyS9Rptq2escYjGj6hqcO5DsPwA?usp=sharing

Topic1: HAR continued…

- Data:
  - 30 subjects' data is randomly split to 70% test and 30% train data
  - Each datapoint corresponds to one of the 6 activities
  - Classes almost balanced
- Baseline:
  - Most feature-based and neural-net based models should easily get > 0.9 $F1_{macro}$
- Dataset paper:
  - https://www.esann.org/sites/default/files/proceedings/legacy/es2013-84.pdf



Topic1: HAR conclusion

- Steps:

1. Get yourself acquainted with the area of human activity recognition
   - Hints: Read blog posts and research papers to get an idea
2. Perform exploratory data analysis
   - Plot both overview statistics and intuitively cherrypicked feature statistics
     - Hints: acceleration should separate walking/sitting etc.
   - Try automatic clustering to hypothesize about problematic classes
     - Hints: T-SNE
3. Train a predictive model of your own choice
   - You need to improve over the naïve baseline in the provided Colab
   - Use **Google Colab**! This is a non-debatable requirement
   - The solution does not need to include a neural network
4. Perform an error analysis
   - Identify easy/hard to predict classes
5. Handover the Colab link and the technical report PDF

Topic2: Food hazard detection

- Classify titles of food-incident reports collected from the web (NLP)
- Data:
  - https://food-hazard-detection-semeval-2025.github.io/
- Baseline:
  - TF-IDF + LinearRegression
    - https://colab.research.google.com/drive/1hv6QifrJ6qRddffoQR1ZQlaWCDBalDhY?usp=sharing
- Leaderboard:
  - https://codalab.lisn.upsaclay.fr/competitions/leaderboard_widget/19955/

"Randsland brand Super Salad Kit recalled due to Listeria monocytogenes"	
hazard:	listeria monocytogenes
hazard-category:	biological
product:	salads
product-category:	fruits and vegetables
"Create Common Good Recalls Jambalaya Products Due To Misbranding and Undeclared Allergens"	
hazard:	milk and products thereof
hazard-category:	allergens
product:	meat preparations
product-category:	meat, egg and dairy products
"Nestlé Prepared Foods Recalls Lean Cuisine Baked Chicken Meal Products Due to Possible Foreign Matter Contamination"	
hazard:	plastic fragment
hazard-category:	foreign bodies
product:	cooked chicken
product-category:	prepared dishes and snacks

Topic3: Plant disease classification

- Train a model to discriminate plant diseases given their images (CV)
- Data:
  - https://github.com/Denisganga/the_plant_doctor/tree/main
- Baseline:
  - https://colab.research.google.com/drive/1o5gXJuB8B-kV0ehfDW1Kv1w0iNHivAKU?usp=sharing

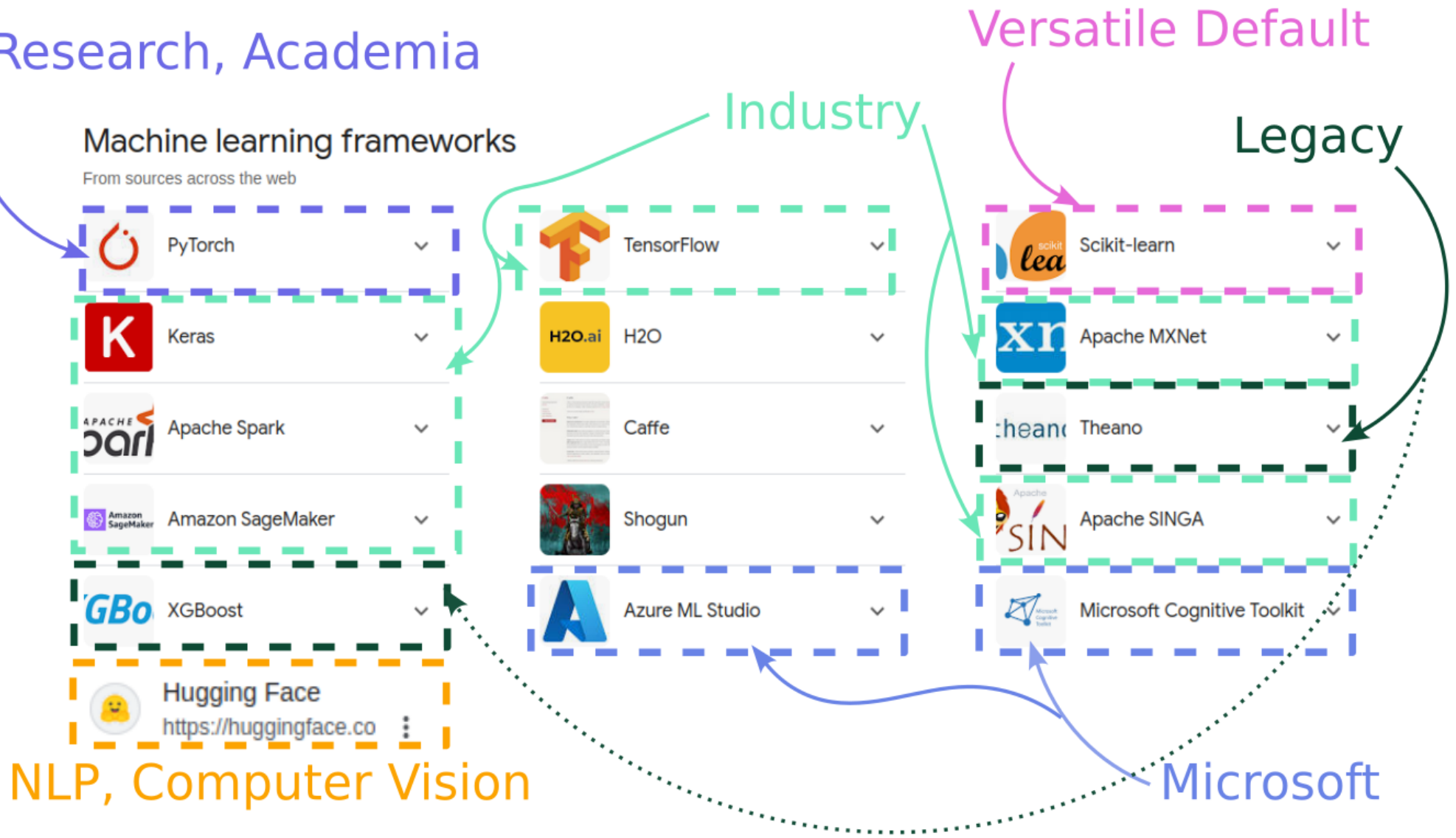
Topic4: Urban sound classification

- Train a model to classify sound recordings on the train split
- Data:
  - UrbanSounds8k: https://www.kaggle.com/code/aadith0/rnn-audio-classification
  - Short sound recordings (e.g., dog bark)
  - 10 classes, slightly imbalanced instances per class
- Baseline:
  - Evaluate the model on the test split
- How to:
  - A naïve solution is to plot the sound recordings and use image classification or raw time-series models
  - Better solutions use signals theory features (e.g. MFCC) or image processing (spectrograms): https://github.com/mashrin/UrbanSound-Spectrogram

Past projects – examples of solutions

- Selected examples from past semesters:
  - AlphaZero for 2-player games
    - https://colab.research.google.com/drive/1l9sGcW466SBNRLsl0KvqVVt4NDJhBShi
  - Spatial temporal prediction
    - https://colab.research.google.com/drive/1LMOP3UqRpRy92mUdfh6iqOGUS781I1y1
  - Feature construction using genetic programming
    - https://colab.research.google.com/drive/1Y-FuI07lnYutbh2rw2SM1NnqJ3ONFujh#scrollTo=2DN7msn0i7ZD
  - Feature hashing
    - https://colab.research.google.com/drive/1KKtwurErcvkEnQfsczCmyJ-PF5DL209C
  - Object recognition with the Vision Transformer
    - https://colab.research.google.com/drive/1_GCpaFtSoRLdW6u7R7Hv-4NshM0Th5rg#scrollTo=XZtQuNSsgYy7

Semestral project – ML frameworks to use



ML frameworks – continued

- Scikit-learn
  - Simple and efficient tools for predictive data analysis
  - Accessible to everybody, and reusable in various contexts
  - Built on NumPy, SciPy, and matplotlib

Warning! Learn about Neural Networks before working with these.

- PyTorch & TensorFlow
  - open-source deep learning frameworks
  - Autograd: dynamic computation graphs
  - GPU Acceleration
- HuggingFace
  - Hub for state-of-the-art pretrained models for NLP & CV
  - Python library

Development tools

- Colab
  + Free GPU, online: no setup & all platforms, seamless graphical interface
  - Time limits, debugging in terminal
- Vim/NeoVim
  + Skill building, easy setup on remote machines
  - Access to machine w. GPU, coding & debugging in terminal, too much fun 
- VS Code
  + GUI, run on remote folders, Copilot
  - Own/access to Machine with GPU, difficulties in setting up remote dev
- PyCharm
  + excellent GUI debugging, very capable IDE, data inspection tools
  - Own/access to Machine with GPU, difficulties in setting up remote dev

Optional development resources

- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). Springer.
  - Well written book on theory with exercises
  - Relevant for data science
- Bishop, C. M., & Bishop, H. (2023). *Deep learning: Foundations and concepts*. Springer Nature.Newer book with focus on neural networks
- Goodfellow, I. (2016). *Deep learning* (Vol. 196). MIT press.
  - Foundational theory behind neural networks
- Jurafsky, M., Speech and language processing
  - https://web.stanford.edu/~jurafsky/slp3/
  - Foundations of NLP with both feature-based and neural-net models
- Kaggle for code and data analysis