Otkrivanje znanja u skupovima podataka

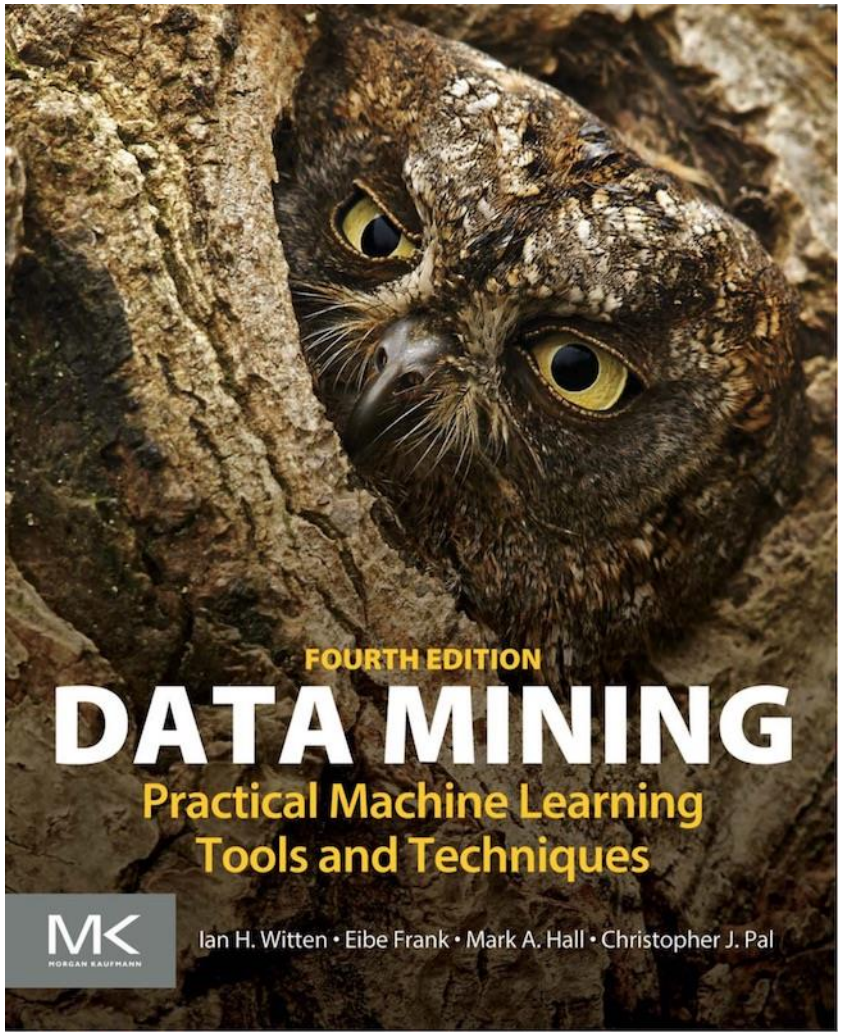
Pripremio:

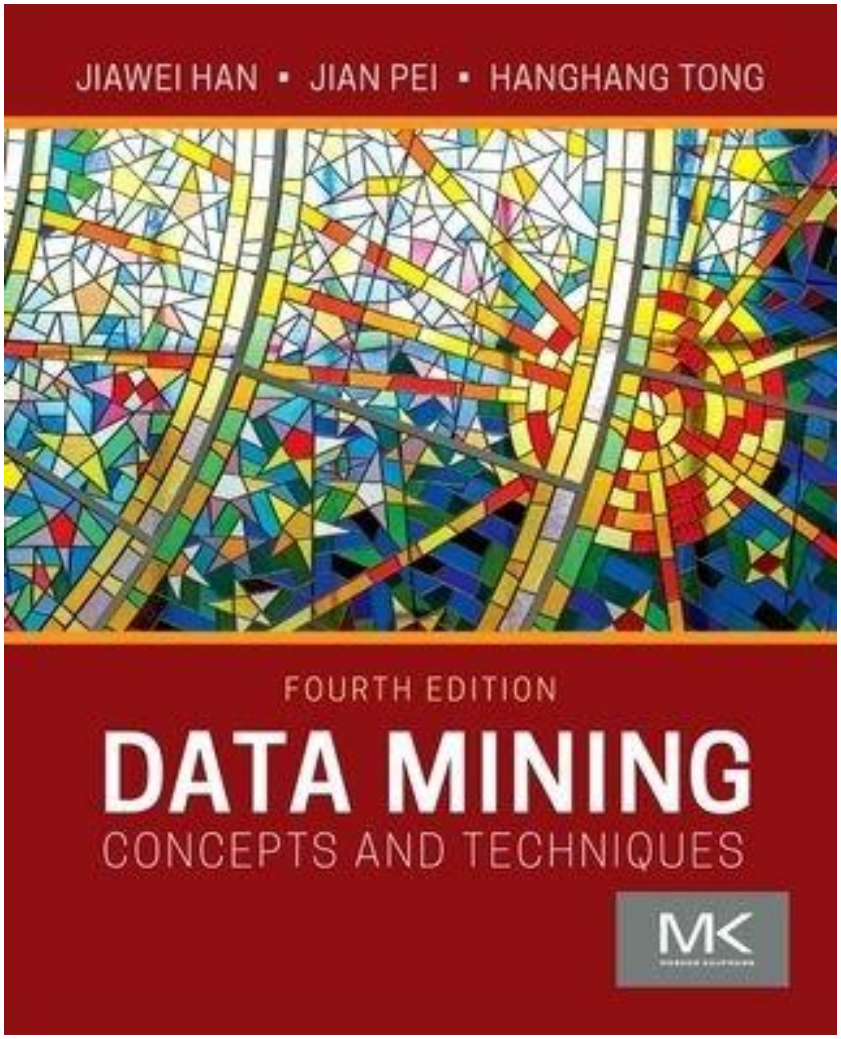
Prof.dr.sc. Nikola Bogunović
Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

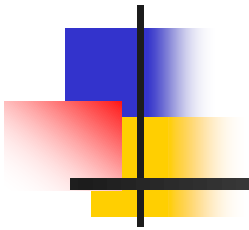
Temeljem izvornih dokumenata (autori zadržavaju sva prava):

- *M.Hall, I.H.Witten, E.Frank, C.J.Pall*
  DATA MINING, Practical Machine Learning Tools and Techniques
  Morgan Kaufmann, 2017., (WEKA 4th. Ed.)
- *Jiawei Han, Micheline Kamber, and Jian Pei*
  Data Mining: Concepts and Techniques, 4th Ed., Elsevier, 2022.
- *T.Michell, MACHINE LEARNING*
  McGraw Hill, 1997
- *Izv.prof.dr.sc. Alan Jović, razne prezentacije*

Otkrivanje znanja u skupovima podataka







Otkrivanje znanja u skupovima podataka

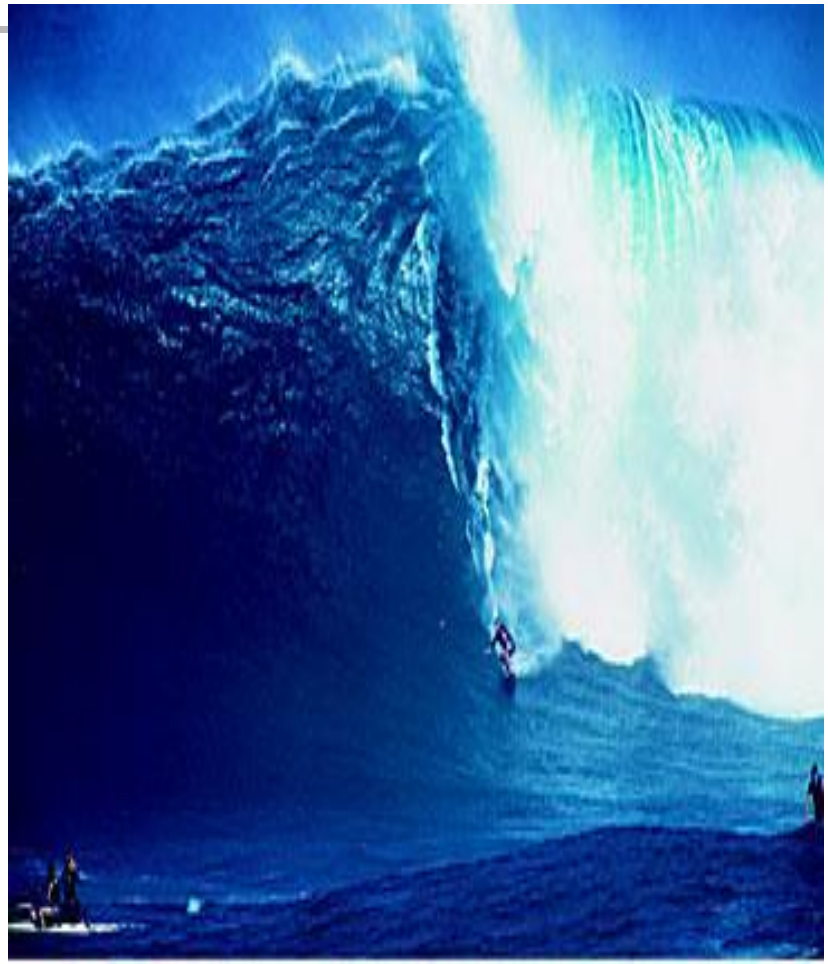
U ovoj prezentaciji:

- Uvodna razmatranja i definicije
- Proces dubinske analize podataka
- Ulazni podaci
- Oblici induciranog znanja

Uvodna razmatranja i definicije

Podatkovni potop (engl. Data Flood)

- Generiranje ogromnih skupova podataka:
  - Bank, telecom, druge poslovne transakcije ...
  - Znanstveni podaci: astronomija, biologija, itd.
  - Web, tekst, i e-poslovanje (e-commerce)
  - . . .



Otkrivanje znanja u skupovima podataka

Zašto želimo analizirati te velike skupove podataka ?

- Utapljamo su u podacima a istivremeno žudimo za znanjem!
- Usko grlo ekspertnog/inženjerskog znanja. Potreba za razvojem sustava koji su složeni ili skupi ako ih razvijamo ručno (potrebno posebno znanje koje nedostaje).
- Potreba za razvojem sustava koji se automatizirano adaptiraju individualnom korisniku (npr. sustavi preporučivanja).
- (Pomoći u razumijevanju procesa učenja kod ljudi i drugih bioloških organizama.)
- Sazrelo je vrijeme (mnogi efikasni algoritmi, dobavljiva velika količina podataka, **osigurani veliki računalni resursi**).

Računalni resursi

FRONTIER

Operational life-time. Lipanj 2022 ->
Computing nodes: AMD 9500 Epyc CPU + 38000 Radeon Instinct GPU (8,335,360 cores)
First „exascale" supercomputer (1.102 exaFLOPS, E=10^{18})
File system: 700 PB, bandwidth: 75 (read), 35 (write) TB/s

Dodatna motivacija

Consumer News and Business Channel

The demand for **data analytic specialists** who know how to manage the tsunami of information, spot patterns within it and draw conclusions and insights is nearing a frenzy.

It's probably the biggest imbalance of supply and demand. The talent pool is, at best, probably 20 percent of the demand.

Qualified big data analysts command impressive salaries. Someone right out of school can earn $150,000, while someone with a year or two of experience and a demonstrated skill can easily make double that. **Job title covers a huge range of disciplines and responsibilities!**

Oprez !!

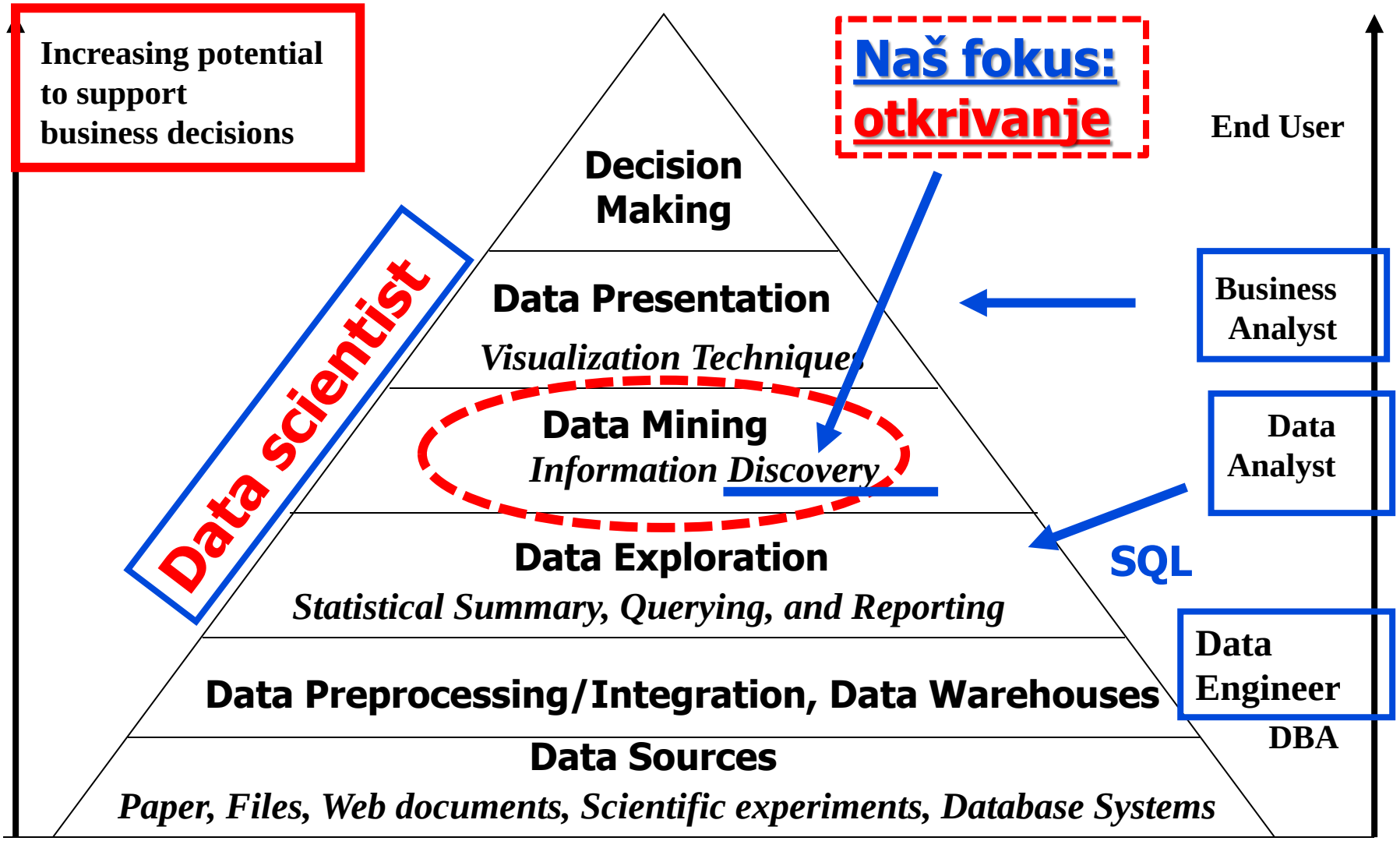
The Washington Post, Nov. 14.2022.

- Over the past week, Silicon Valley companies have laid off 20,000 employees.
- Twitter, Facebook parent Meta, payment platform Stripe, software service firm Salesforce, ride-hailing company Lyft and a growing list of smaller companies all laid off double-digit percentages of their workers.
- Google and Amazon have recently instated hiring slowdowns and freezes.
- The layoffs come just a year after Silicon Valley was at its peak, with valuations of Big Tech companies spilling into the trillions.
- The bull market of the past decade — which created massive amounts of wealth for tech investors, workers and the broader economy — is decidedly over.
- Most analysts are referring to the turn-of-the-century dot-com crash.
- Possible reason: inflation, energy shocks, higher interest rates, reduced investment budgets, and sparser start-up funding.

Evolution of Sciences to Data science

- Before 1600, **empirical science**
- 1600-1950s, **theoretical science**
  - Each discipline has grown a *theoretical* component. Theoretical **models** often motivate experiments and generalize our understanding.
- 1950s-1990s, **computational science**
  - Over the last 50 years, most disciplines have grown a third, *computational* branch .Computational Science traditionally meant **simulation**. It grew out of our **inability to find closed-form solutions for complex mathematical models.**
- 1990-now, ***data science***
  - The flood of data from new scientific instruments and simulations
  - The ability to economically store and manage petabytes of data online
  - The Internet and computing Grid makes archives universally accessible
  - Scientific info. management, acquisition, organization, query, and visualization tasks scale almost linearly with data volumes. ***Data mining*** is a **major new challenge**!

Data Science in Business Intelligence



Otkrivanje znanja u skupovima podataka

Data Mining = Dubinska analiza podataka

To je: ***Proces polu-automatizirane analize*** velikih skupova podataka s ciljem ***pronalaženja (otkrivanja) obrazaca (uzoraka, zakonitosti)*** koji su:

- **Valjani** - vrijede za **nove** podatke s nekom izvjesnosti
- **Neočigledni** - ne trivijalni, nenadani
- **Korisni** (engl. *Actionable*) - omogućuju postupanje i uporabu
- **Razumljivi** – moguće ih je interpretirati

Otkrivanje znanja u skupovima podataka

Dubinska analiza podataka (a.k.a. Data mining)

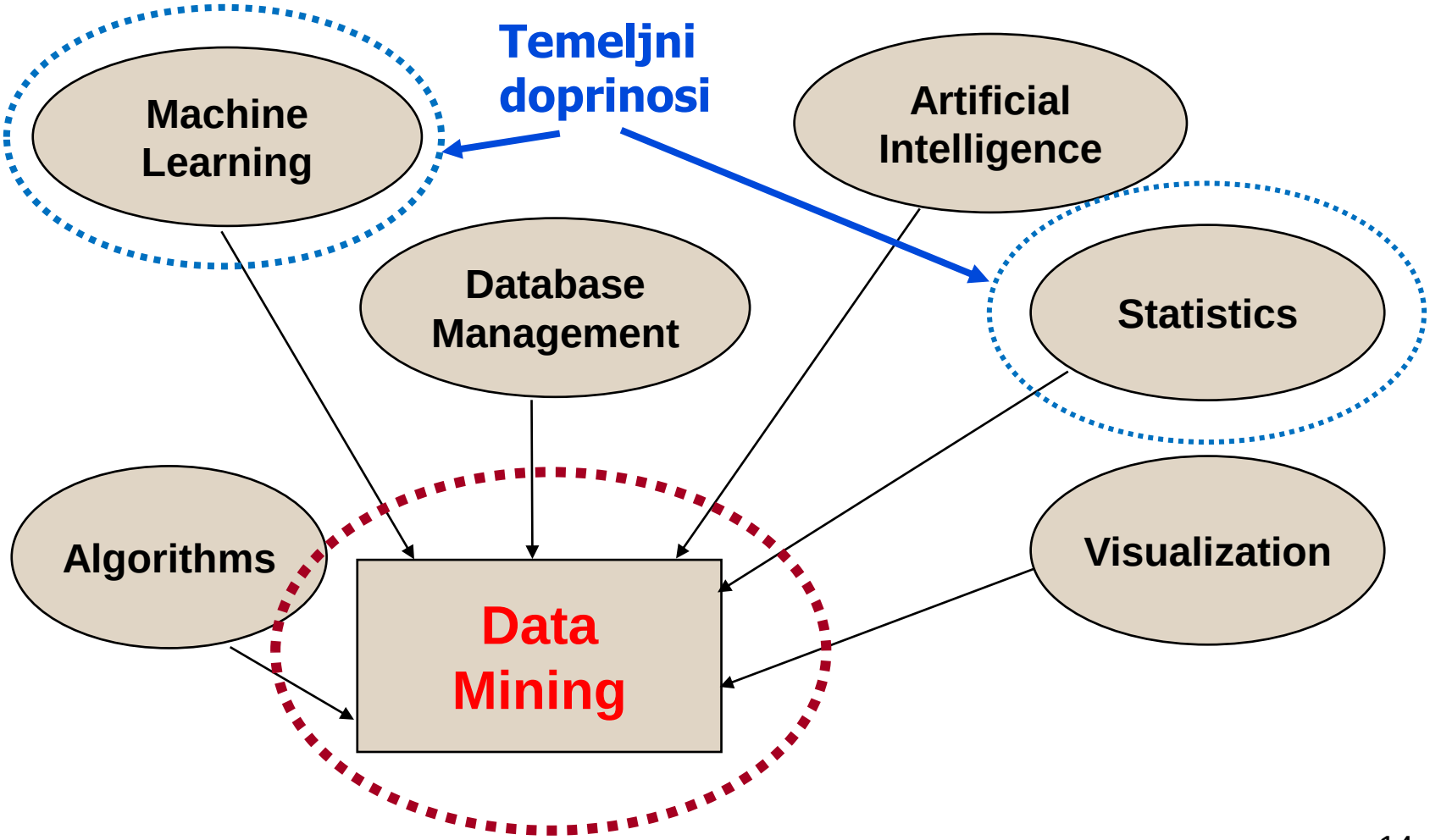
Alternativni nazivi:

- Otkrivanje znanja u bazama podataka
- Izlučivanje znanja
- Analiza podataka/uzoraka
- Arheologija podataka
- …

Što nije dubinska analiza podataka:

- Jednostavno pretraživanje baza podataka (npr. SQL)
- (Deduktivni) ekspertni sustavi

Integracija mnogih tehnologija



Odnos statistike, strojnog učenja i dubinske analize podataka

Statistika:

- Većim dijelom teorijski zasnovana
- Fokusira se na **testiranje postavljene hipoteze** ali ne samo to.

Strojno učenje:

- Fokusira se na **poboljšanje performansi agenta koji uči**
- Dodatna primjena u robotici i **učenju u stvarnom vremenu** (područja koja nisu dio dubinske analize podataka u užem smislu)

Dubinska analiza podataka i otkrivanje znanja:

- ***Integrira*** teoriju i heuristiku
- Fokusira se na **cjelokupni proces otkrivanja znanja** uključujući čišćenje podataka, učenje, integraciju i vizualizaciju rezultata
- **Generira hipotezu !!!!**

Nema izrazite podjele ni granice, prihvaćamo kontinuum.

Odnos strojnog učenja i dubinske analize podataka

- Dubinsku analizu podataka izvode ljudi s ciljem pronalaženja zanimljivih uzoraka i obrazaca u skupu podataka.
- Dubinska analiza podataka primjenjuje tehnike razvijene u području strojnog učenja.
- A.Samuel (1959): Strojno učenje daje računalima sposobnost da uče bez da su eksplicitno programirani.
- Strojno učenje temeljem informacija o odnosima između elemenata u skupu podataka stvara modele kako bi se predvidjeli budući rezultati.
- Algoritmi strojnog učenja se poboljšavaju čestim prihvatom ulaznih podataka za učenje (treniranje).
- The main and most important difference between data mining and machine learning is that without the involvement of humans, data mining can't work, but in the case of machine learning human effort only involves at the time when the algorithm is defined after that it will conclude everything on its own.

Tro-dimenzijski pogled na dubinsku analizu podataka

- **Različiti izvori podataka**
  - Baze podataka (proširene-relacijske, objektno usmjerene, heterogene, povijesne – „legacy”, skladišta podataka, transakcijski podaci, nizovi, prostorno-vremenski podaci, vremenski nizovi, sekvence, tekst i web, multimedija, grafovi, društvene i informacijske mreže
- **Različiti oblici izlučenog znanja**
  - Karakterizacija, diskriminacija, pridruživanje (asocijacija), klasifikacija, grupiranje, analiza trenda i devijacija, analiza nepripadanja (engl. *Outlier analysis*) itd.
  - Opisna ili prediktivna analiza.
  - Višestruke funkcije i analiza na više razina.
- **Različite tehnike otkrivanja znanja**
  - Brzi odgovori na višedimenzijske upite (OLAP), strojno učenje,, statistika, raspoznavanje uzoraka, vizualizacija, itd.

Neki česti tipovi analize

Generalizacija i diskriminacija

- Poopćavanje i sažimanje te usporedba i razlikovanje skupova podataka.
- Višedimenzijski opis nekog koncepta.
  - Npr. Kišna i suha zemljopisna područja.

Pridruživanje (asocijacija) i korelacijska analiza te analiza uzroka i posljedica

- Koje stvari obično kupujete zajedno u vašem omiljenom supermarketu ?
- *Ako pelene tada i pivo [0.5%, 75%]* - potpora, uvjerenost
- Kako otkriti ovakve uzorke u velikim skupovima podataka ?
- Kako iskoristiti ovakve uzorke za klasifikaciju, grupiranje i druge primjene ?

Neki česti tipovi analize

Klasifikacija

- Oblikovati modele (funkcije) na temelju podataka za učenje.
- Opisati i razlikovati razrede ili koncepte za **buduću predikciju**.
- Npr.: klasificirati države na temelju klime, detekcija kriminalnog poslovanja, izravno oglašavanje, klasifikacija zvijezda, bolesti, web stranica, …

Grupiranje

- Grupirati podatke tako da tvore nove kategorije.
- **Nenadgledano učenje**. Ciljani razredi su nepoznati.
  - Npr.: grupiranje kuća za stanovanje da se uoče uzorci razdiobe.

Neki česti tipovi analize

Analiza nepripadanja

(enl. *Outlier analysis*)

Outlier: Podatkovni objekt koji se ne podudara s općim zakonitostima unutar nekog skupa podataka.

Npr.: Detekcija rijetkih događaja, kriminalnih radnji i sl.

Vrijeme i sekvencije

Trendovi, vremenski nizovi, slijed/sekvencije uzoraka, analiza periodiciteta, analiza bioloških sekvenci, potencijalno beskonačni nizovi podataka…

Npr.: Prvo se kupi digitalna kamera a zatim SD memorija velikog kapaciteta, analiza DNK sekvenci, analiza srčanog ritma…

Neki česti tipovi analize

Analiza raznih struktura i informacijskih mreža

- Dubinska analiza grafova (otkrivanje čestih podgrafova, XML stabala, web fragmenata, …)

- Analiza informacijskih mreža

Društvene mreže (aktori i odnosi)

Višestruke heterogene mreže (jedna osoba na više mreža)

Analiza poveznica (engl. Links analysis)

- Analiza web-a

Analiza uporabe, otkrivanje mišljenja (izbori ?), otkrivanje zajednica na mreži, …

Posebni problemi u dubinskoj analizi podataka

Metodologija dubinske analize

- Analiza različitih i novih vrsta znanja (heterogenost)
- Dubinska analiza u višedimenzijskom prostoru
- Interdisciplinaran napor
- Otkrivanje znanja u mrežnom okruženju
- Rukovanje šumom, neizvjesnim i nekompletnim podacima
- Evaluacija uzoraka

Interakcija korisnika

- Interaktivna dubinska analiza
- Uvođenje pozadinskog (engl. *Background*) znanja
- Prezentacija i vizualizacija rezultata dubinske analize

Posebni problemi u dubinskoj analizi podataka

Efikasnost i skalabilnost

- Fokus na algoritmima dubinske analize
- Paralelne, raspodijeljene, inkrementalne metode, beskonačni nizovi

Velika raznolikost tipova podataka (ne samo izvora)

- Rukovanje složenim tipovima podataka
- Dubinska analiza dinamičkih, umreženih i globalnih repozitorija podataka

Društveni aspekti dubinske analize podataka

- Društveni utjecaj dubinske analize podataka
- Očuvanje privatnosti
- Nevidljiva dubinska analiza podataka

Privatnost podataka - 1

Narušavanje privatnosti (primjeri iz USA)**:**

- DoubleClick
  - Povezivanje obrazaca korištenja Interneta s imenima, tel. brojevima, adresama i drugim demografskim podacima.
  - Tvrtka je izjavila da podaci neće biti javno dostupni.
  - Poslije niza sudskih tužbi tvrtka je odustala od projekta.
- Naviant
  - Prodaja drugim tvrtkama podatke iz registracijskih kartica pri kupnji proizvoda.
- USWest
  - Koristila je zapise o telefonskim razgovorima za marketing.
- Trans Union
  - Podatke o povijesti otplata kredita prodala drugima.
- Ljudski genom
  - Povezivanje individualne genetičke strukture s potencijalnim bolestima, neke tvrtke mogu iskoristiti za promjenu police zdravstvenog osiguranja svojih zaposlenika.

Privatnost podataka - 2

Slučaj Facebook i Cambridge Analytica

- Aleksandr Kogan, data scientist pri University of Cambridge unajmljen je od tvrtke Cambridge Analytica da razvije aplikaciju „This is your digital life”, izvorno samo za akademsku uporabu.
- Nekoliko tisuća korisnika Facebook-a pristalo je za određenu naknadu odgovoriti na skup pitanja iz ove aplikacije.
- Odgovori opisuju psihološki profil korisnika.
- 2010. god. Facebook je dozvolio da ova aplikacija prikuplja osobne podatke ne samo od izvornih korisnika koji su pristali nego i o njihovim svim Facebook prijateljima.
- Cambridge Analytica je tako prikupila osobne podatke više miliona Facebook korisnika, najvećim dijelom za političko oglašavanje.
- 2016. god. Cambridge Analytica pružila je uslugu analize podataka u predsjedničkoj kampanji D. Trampa, a utemeljeno se vjeruje u utjecaj na referendum o Brexitu.
- Nakon otkrića (Guardian,, New York Times), 2018, god. Cambridge Analytica je bankrotirala a Facebook novčano kažnjen (?).

FER – Kolegij: "Otkrivanje znanja u skupovima podataka"

Kolegij obuhvaća numeričke i simboličke postupke **dubinske analize** i otkrivanja strukturnih uzoraka u podacima i signalima. **Nije posebno usmjeren na ogromne skupove.**

Sadržaj kolegija:

- **Proces** dubinske analize podataka (definicije koncepata kao ciljnih funkcija, primjera i značajki/atributa podataka).
- **O ulaznim podacima** (prikupljanje i priprema za analizu)
- **Predstavljanje otkrivenog znanja -koncepta** (tablice i stabla odlučivanja, razredbena pravila i pravila pridruživanja, skupine i drugo).
- **(Algoritmi za indukciju znanja.)**
- **Evaluacija rezultata**.
- **Primjena** postupaka strojnog učenja u poslovnom odlučivanju, financijama, tehnici i medicini.

Primjer kolegija sa Stanford University

(Kolegij na FER-u: Analiza velikih skupova podataka)

Mining Massive Data Sets (CS 246) 2021

Neke nastavne cjeline:

- DFS and Map-Reduce, Spark (Apache data flow)
- Frequent Itemsets Mining
- Clustering
- Dimensionality Reduction
- PageRank
- Social Networks
- Algorithms on Large Graphs
- Recommender Systems
- Mining Data Streams
- Computational Advertising

Book: Mining of Massive Datasets

Jure Leskovec, Anand Rajaraman, Jeff Ullman

Download full version of the book.

FER – Kolegij: "Otkrivanje znanja u skupovima podataka"

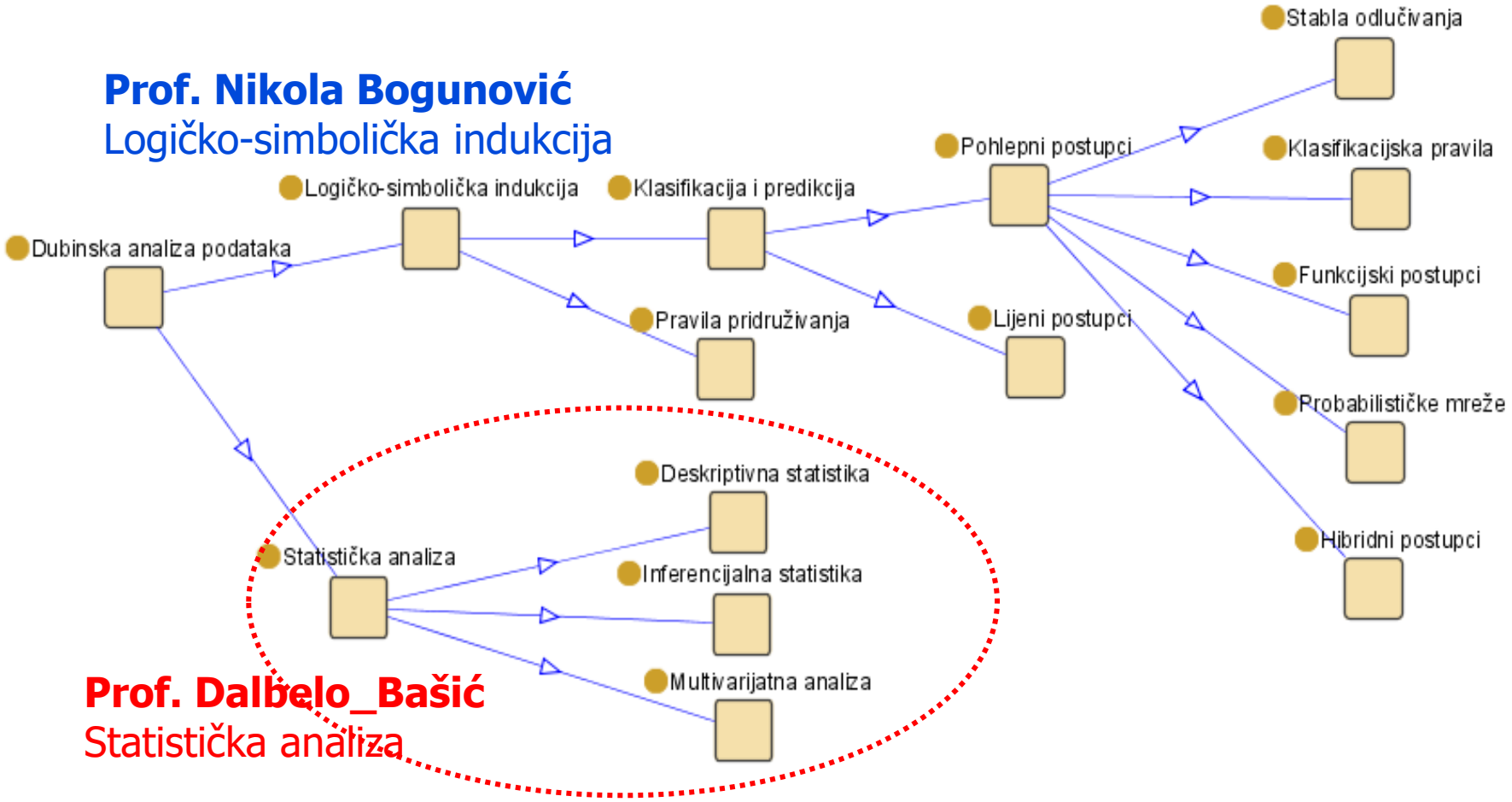
Očekivana prethodna znanja:

- **Matematika:**
  - **Matematička logika**
  - **Statistika**
  - **Linearna algebra (matrice)**
  - **Diferencijalni račun**

- **Vještine programiranja:**
  - **Python**
  - **R**
  - **Java**

Otkrivanje znanja u skupovima podataka

Jedna neformalna klasifikacija dubinske analize podataka



Otkrivanje znanja u skupovima podataka

FER - Kolegij "Otkrivanje znanja u skupovima podataka"

Cilj kolegija: studenti trebaju steći znanja i vještine za rješavanje srednje do teških problema iz dubinske analize podataka. Rezultati dubinske analize moraju biti **neočekivani** i **uporabivi** (engl. *actionable*).

Provjera ostvarenja cilja: samostalna dubinska analiza podataka u obliku **seminarskog rada ili istraživačkog članka** na kraju semestra (**vidi primjerke na web stranici predmeta**).

Nastavni materijali (vidi web stranicu kolegija):

Otkrivanje znanja u skupovima podataka

Sadržaj web stranice predmeta:
http://www.zemris.fer.hr/predmeti/kdisc/ref1.html

1. Predavanja (PowerPoint ili PDF)
2. Pregled područja (dokument s Instituta R.Bošković)
3. Prikaz samo nekih algoritama
4. Poveznica na WEKA sustav za analizu podataka
5. Poveznica na RapidMiner sustav za analizu podataka
6. Članak o 10 najčešćih algoritama
7. Poveznice na arhive skupova podataka (KDnuggets, UCI, Kaggle Competitions)
8. Primjeri seminarskih radova

Kaggle Competitions

(https://www.kaggle.com/competitions)

Neki primjeri:

- **Titanic**: Temeljem podataka o putnicima treba odgovoriti na pitanje: „Koja vrsta ljudi ima(la) najveće šanse za preživljavanje?” Kontinuirano natjecanje.

Neka tekuća natjecanja (2022.g.) s rokovima više mjeseci:

- **DFL - Bundesliga** Data Shootout
  Identify plays based upon video footage
- **American Express** - Default Prediction
  Predict if a customer will default in the future
- **March Machine Learning Mania 2022** - Men’s
  Predict the 2022 College Men's Basketball Tournament
- **JPX Tokyo Stock Exchange** Prediction
  Explore the Tokyo market with your data science skills

Otkrivanje znanja u skupovima podataka

Struktura seminarskog rada:

1. **Uvod (o domeni i podacima)**
2. **Ciljevi dubinske analize i kako ih prikazati (vrlo važno i obavezno poglavlje)**
   - **Jasno izraziti u nekoliko rečenica**
3. **Dohvat i priprema podataka**
4. **Provođenje analize**
   - **2 do 3 postupka**
   - **Posebnu pažnju obratiti na prilagodbu podataka postupcima**
   - **Navesti dobivene mjere/ocjene točnosti i efikasnosti analize**
5. **Diskusija rezultata analize**
   - **U kojoj mjeri su dostignuti ciljevi analize**

Otkrivanje znanja u skupovima podataka

FER - Kolegij "Otkrivanje znanja u skupovima podataka"

Statistički pristupi dubinskoj analizi podataka

(prof.dr.sc. B.Dalbelo-Bašić)

- **Opisna statistika**
  - **Direktno mjerenje parametara populacije**
- **Inferencijalna statistika**
  - **Mjerenja statistika uzorka, hipoteze, intervali pouzdanosti**
- **Multivarijatna statistička analiza**
  - **Više postupaka modeliranja podataka, u ovisnosti o prirodi varijabli i tipu problema**

Korisna knjiga: (Predložio N.Bogunović)
Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani:
An Introduction to Statistical Learning, with Applications in R
University of Southern California (download the book PDF).

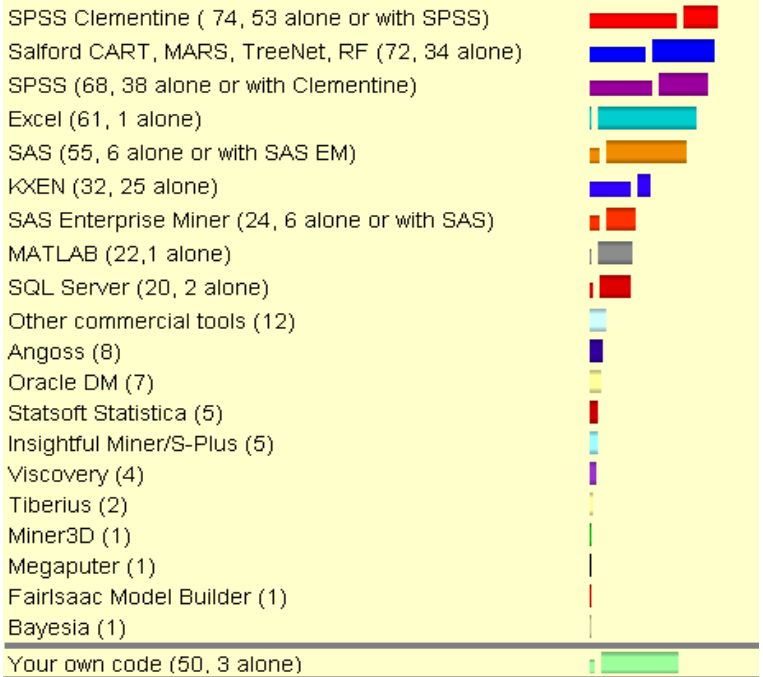
Otkrivanje znanja – **alati** i primjene

Otkrivanje znanja – alati i primjene

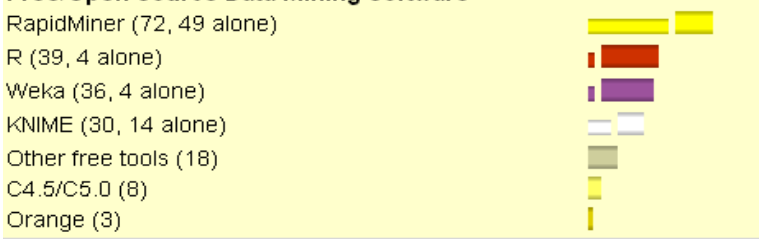
What data mining tools have you used for a real project (not just f in the past 6 months? [347 voters]

For tools with 20 votes or more, we split results into "alone" votes - where alone (narrow bar), and the second (wide) bar to the number votes where select as one among several; Tools are ordered in descending order of t votes

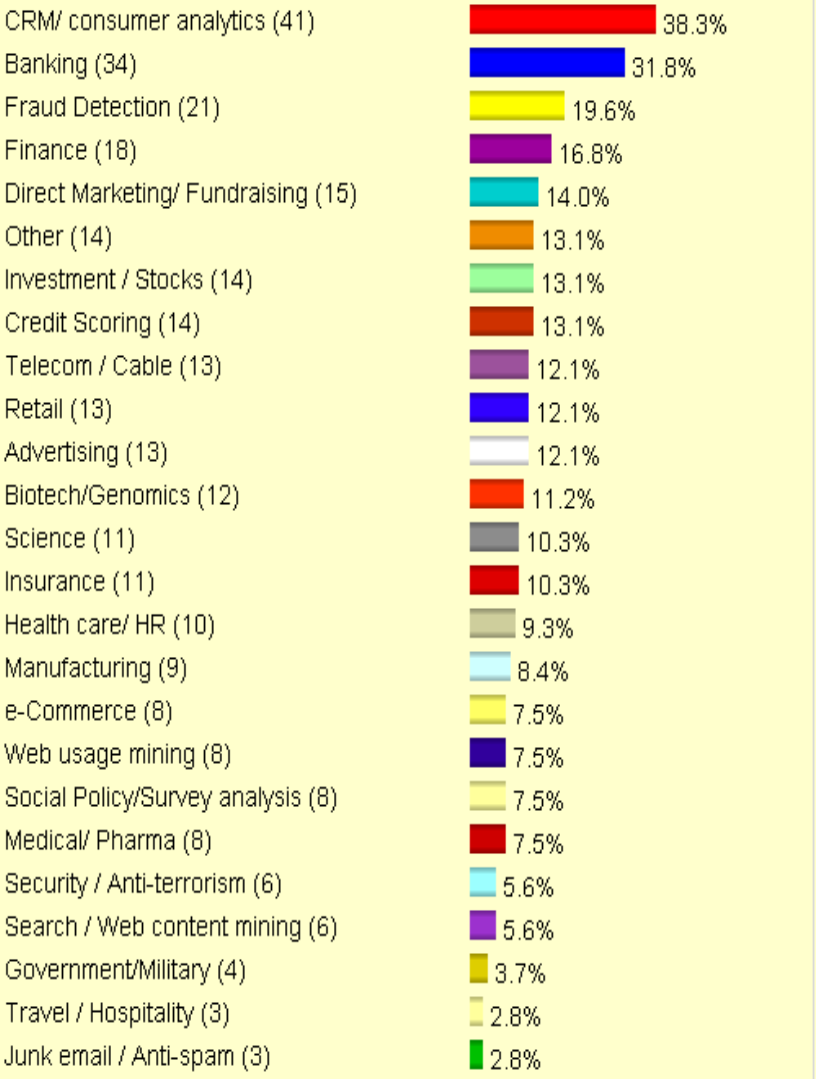
Commercial Software



Free/Open Source Data Mining Software



Industries / Fields where you applied Data Mining in 2008: [107 voters]



Otkrivanje znanja – alati i primjene

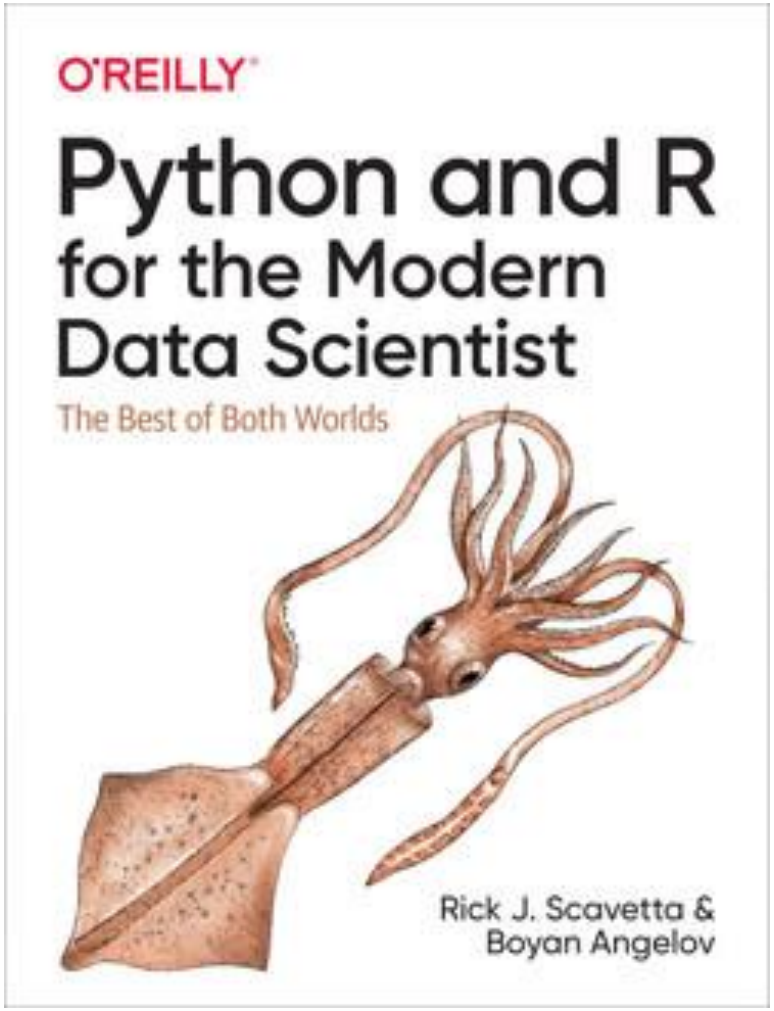
PYTHON PACKAGES FOR DATA MINING

- **NUMPY - fundamental package for scientific computing with Python.**
- **SCIPY - software for mathematics, science, and engineering.**
- **PANDAS - fast, flexible, and expressive data structures designed to make working with "relational" data. For doing practical, real world data analysis in Python.**
- **MATPLOTLIB - API for embedding plots into applications.**
- **IPYTHON - command shell for interactive computing.**
- **SCIKIT-LEARN – library for machine learning in Python.**

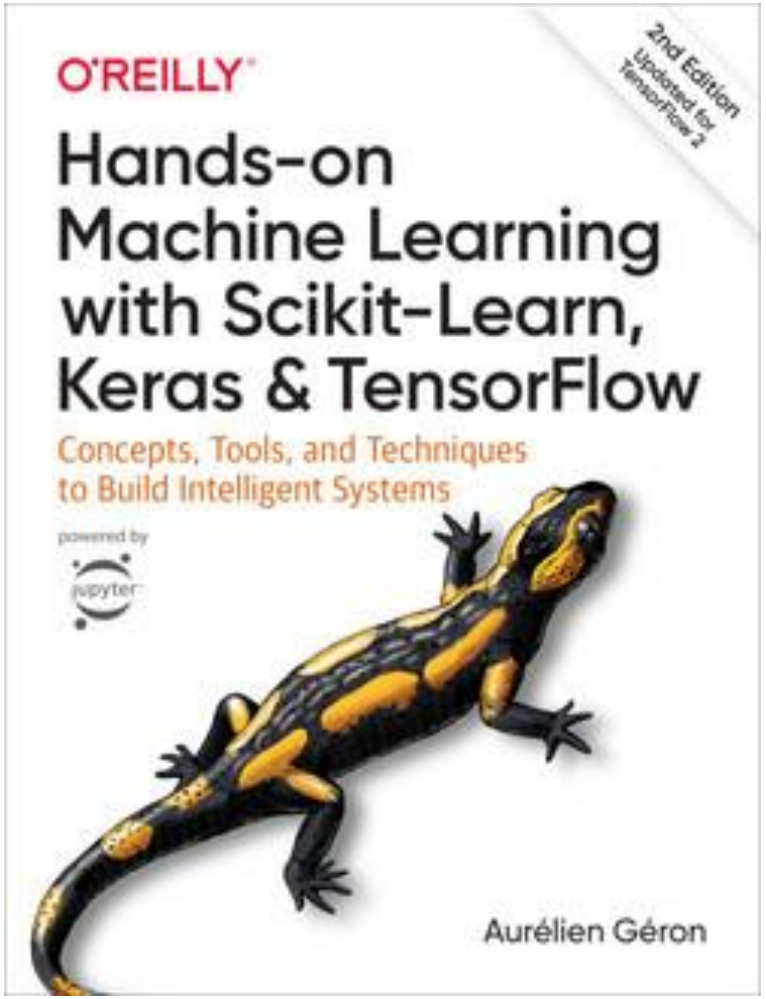
R – programming environment for data analysis and graphics

- **an effective data handling and storage facility,**
- **a suite of operators for calculations on arrays, in particular matrices,**
- **a large, coherent, integrated collection of intermediate tools for data analysis,**
- **graphical facilities for data analysis and display**
- **a simple and effective programming language (S - izrazi) which includes conditionals, loops, user defined recursive functions and input and output facilities.**

Otkrivanje znanja – alati i primjene



2021. god.



2019. god.

Razvojne okoline (a.k.a. IDE)

Jupyter

- Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.
- Jupyter is 100% open-source software, free for all to use.
- While Jupyter runs code in many programming languages, Python is a requirement (Python 3.3 or greater, or Python 2.7) for installing the Jupyter Notebook.

PyCharm

- PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactorings and rich navigation capabilities.
- In addition to Python, PyCharm supports JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, template languages, AngularJS, Node.js, and more.

Razvojne okoline (a.k.a. IDE)

Spyder

- Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts.
- It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, and beautiful visualization capabilities of a scientific package.

Anaconda distribution (Python 3.9 distribution + Spyder)

- Open-source to perform Python/R data science and machine learning
- Anaconda is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 1,500+ open source packages.
- Anaconda is free and easy to install

Prikupljanje podataka s weba

Python usmjereno

Srapy

- An open source and collaborative framework for extracting the data you need **from websites**.
- Scrapy runs on Python 2.7 and Python 3.5 or above under CPython (default Python implementation) and PyPy (starting with PyPy 5.9).
- Scrapy is written in pure Python and depends on a few key Python packages (among others):
  - lxml, an efficient XML and HTML parser
  - parsel, an HTML/XML extraction library written on top of lxml,
  - w3lib, a multi-purpose helper for dealing with URLs and web page encodings
  - twisted, an asynchronous networking framework
  - cryptography and pyOpenSSL, to deal with various network-level security needs

Tutorial: Scraping famous quotes

Prikupljanje podataka s weba

Python usmjereno

Beautiful Soup

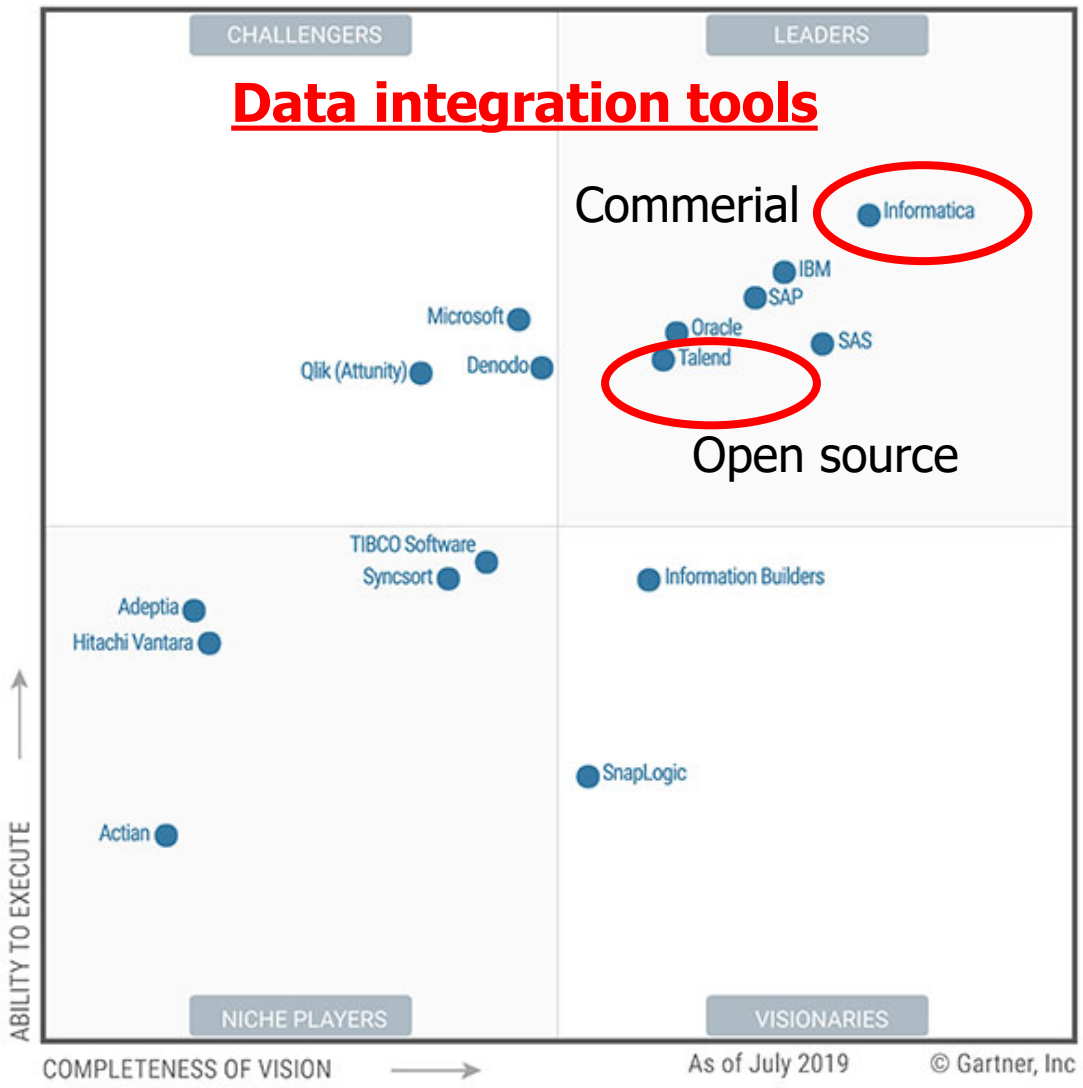
- Beautiful Soup is a library that makes it easy to scrape information **from web pages**. It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.
- Beautiful Soup development will exclusively target Python 3
- Beautiful Soup automatically converts incoming documents to Unicode and outgoing documents to UTF-8. You don't have to think about encodings, unless the document doesn't specify an encoding and Beautiful Soup can't detect one. Then you just have to specify the original encoding.
- Beautiful Soup sits on top of popular Python parsers like **lxml** and **html5lib**, allowing you to try out different parsing strategies or trade speed for flexibility.

Integracija podataka iz baza - ETL

ETL (Extract, Transform, and Load) Process

ETL is defined as a process that extracts the data from different RDBMS source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system.

Figure 1. Magic Quadrant for Data Integration Tools



Source: Gartner (August 2019)

Integracija podataka iz baza - ETL

Talend

- Open Studio for Data Integration Features.
- Free open source Apache license.
- Open source distribution has limited functionality.
- Graphical design environment.
- RDBMS: Oracle, Teradata, Microsoft SQL server, and more
- File management: open, move, compress, decompress without scripting.
- Control and orchestrate data flows and data integrations with master jobs.
- Map, aggregate, sort, enrich, and merge data.

Alati za vizualizaciju podataka

Tableau

- Tableau is a Data Visualization tool that is widely used for Business Intelligence but is not limited to it.
- It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights.
- Tableau Public (server) does not require any licence.
- Comes with a limitation that all data and workbooks are made public to all Tableau users.
- Prerequisites:
  - Awareness of all the types of graphs such as bar graph, line charts, histograms.
  - Basic understanding of database management (datatypes, joins, drill down, drill up etc)

Alati za vizualizaciju podataka

Microsoft Power BI

- Jednostavniji od Excela i Tableau-a.
- Napredna analitika i izvještavanje, obradu informacija i njihov prikaz kroz niz nadzornih ploča (dashboards) koji se jednostavno mogu prilagođavati specifičnim potrebama korisnika.
- Sam alat dostupan je u cloud, desktop i mobile verziji s mogućnošću povezivanja s nizom vanjskih sustava.
- Ako korisnik posjeduje CRM sustav, centralni repozitorij za spremanje dokumenata (npr. izvještaja u Excelu ili bilo kakve baze) ili koristi servise kao što su Google Analytics, Facebook i druge, Power BI omogućuje jednostavno povezivanje s njima te prikaz podataka iz tih sustava na jednom centralnom mjestu i na standardizirani način.
- Desktop verzija je slobodna.

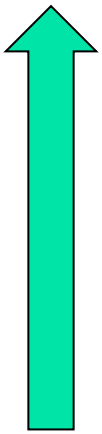
Alati za vizualizaciju podataka

QlikView

- QlikView is a Business Intelligence (BI) data discovery product for creating guided analytics applications and dashboards tailor-made for business challenges.
- The software enables user to uncover data insights and relationships across various sources with QlikView's Associative Data Indexing Engine.
- QlikView can connect to most of the popular databases like MySQL, SQL Server, Oracle, Postgress etc. It can fetch data and table structures into QlikView environment and store the results in its memory for further analysis.
- It can also connect with **R** through API integration.
- QlikView Personal Edition license is free to create QlikView documents for personal use.

Uloga strojnog učenja u otkrivanju znanja u skupovima podataka

Vertikalno povezivanje disciplina



- **Otkrivanje znanja u skupovima podataka**
  zasnovano je na procesu
- **Dubinske analize podataka** (engl. ***Data mining***)
  koja se temelji na postupcima (algoritmima)
  strojnog učenja (engl. ***Machine learning***)

Kako je strojno učenje u temelju otkrivanja znanja u skupovima podataka potrebno je u uvodu navesti poveznicu sa strojnim učenjem.

Strojno učenje (engl. *machine learning*)

- **Herbert Simon** (1978.g. Nobelova nagrada u području ekonomije – „Decision making in economic organizations”)**: “Strojno učenje je proces tijekom kojega sustav (agent) automatizirano poboljšava performanse temeljem iskustva”.**

- **Temeljni entiteti u procesu strojnog učenja:**
  zadatak (**T**), **iskustvo** (**E**), metrika **performansi** (**P**)

- **Kako se mjere performanse?**
  Ovisi o domeni primjene !

Strojno učenje

Primjeri procesa strojnog učenja:

Poboljšati zadatak **T (task)** u odnosu na metriku **performanse P**, temeljeno na iskustvu **E (experience)**

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

Deep Mind (Google): AlphaGo, AlphaGoZero, AlphaZero
T: Playing chess, shogi, go
P: Percentage of wins
E: Only rules, „plays against itself".

Strojno učenje

Neki karakteristični oblici strojnog učenja:

- **Nadzirano učenje** (engl. *supervised learning*)

  Za dani skup primjera ili ulaza s pripadajućim poznatim kategorijama ili izlazima, predvidjeti kategorije ili izlaze budućih primjera ili ulaza (npr. klasifikacija, regresija, …).

- **Nenadzirano učenje** (engl. *unsupervised learning*)

  Za dani skup primjera ili ulaza automatizirano otkrivanje reprezentacije (predstavljanja), strukture ili značajki (npr. grupiranje, sažimanje podataka, detekcija nepripadajućih vrijednosti, …).

- **Učenje s povratnom vezom** (engl. *reinforcement learning*):

  Nagrada ukoliko akcije predviđene modelom daju uspjeha, odnosno **kazna** ako ne daju. Učenje sekvenci akcija koje maksimiziraju očekivanu nagradu (vidi DeepMind: AlphaGo, AlphaZero)

  - **Duboko učenje** (umjetne neuronske mreže s više slojeva)

Strojno učenje

Oblikovanje sustava strojnog učenja:

- Odaberi iskustvene podatke za učenje (E).
- Definiraj precizno što se želi naučiti, t.j. definiraj **ciljnu funkciju** (engl. *target function*) ili **ciljni koncept**.
- Odaberi **kako predstaviti ciljnu funkciju/koncept**.
- Odaberi postupak ili postupke (algoritme) koji će izlučiti **aproksimiranu ciljnu funkciju** (ili **koncept**) iz iskustvenih podataka.

Npr. za igru dame ; šaha, GO, i sl.:

Ciljna funkcija treba generirati najbolji potez za dano stanje na ploči i skup dozvoljenih poteza.

```
ChooseMove(board, legal-moves) → best-move
```

možda ?

Strojno učenje

Predstavljanje ciljne funkcije ili koncepta

- Ciljna funkcija ili koncept može se predstaviti na mnogo načina.: "look-up" tablica, simbolička pravila, numerička funkcija, … .
- Postoji kompromis između izražajnosti (ekspresivnosti) i lakoće učenja ciljne funkcije ili koncepta.
- Što je predstavljanje funkcije (koncepta) izražajnije to će moći bolje aproksimirati proizvoljnu funkciju (koncept) ali će biti potrebno više iskustvenih primjera za učenje.

Postupak (algoritam) učenja

- Uporabom iskustvenih primjera **izluči jednu** hipotezu iz prostora hipoteza (svih funkcija/koncepata) koja najbolje opisuje ciljnu funkciju i nadati se da će uspješno moći primijeniti (generalizirati) na nove (neviđene) primjere.
- Pri tome se minimizira neka mjera pogreške (funkcija gubitka – *loss function*) i maksimizira mjera uspješnosti.

Strojno učenje

Izlučivanje ciljne funkcije ili koncepta

- Izlučivanje i aproksimacija ciljne funkcije može se promatrati kao **pretraživanje prostora skupa hipoteza** (reprezentacija mnogih funkcija) za onom hipotezom (funkcijom) koja najbolje opisuje iskustvene primjere kroz ciljnu funkciju.
- Različite metode učenja pretpostavljaju različite prostore hipoteza (jezike za predstavljanje) i pritom koriste različite tehnike pretraživanja.
- Neki oblici predstavljanja ciljne funkcije ili koncepta:
  - Numerička (npr. linearna regresija, potporni vektori)
  - Simbolička (npr. pravila, stabla odlučivanja)
  - Funkcija temeljena na pohranjenim primjerima(npr. najbliži susjedi)
  - Probabilistički modeli(npr. Bayes, Markov)
- Neki algoritmi pretraživanja prostora:

  smanjenje gradijenta, dinamičko programiranje, podijeli i vladaj, evolucijski, … .

Dubinska analiza podataka

Učenje (generalizacija)
kao postupak **pretraživanja**

Aspekt strojnog učenja koji dubinsku analizu podataka **razlikuje od statističkog pristupa**

Učenje (generalizacija) kao **pretraživanje**

Vrlo jednostavan i **nerealan** primjer:

Temeljem prošlih igara (**E** - **iskustvo**) odredi da li igrati ili ne tenis (**T** - **zadatak**) za bilo koji skup atributa.

Atributi (značajke) i njihove vrijednosti — Ciljni atribut

Prošle igre, primjeri, **iskustvo**

(enl. Experience)

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne
Oblačno	Hladno	Normalna	Da	Da
Sunčano	Blaga	Visoka	Ne	Ne
Sunčano	Hladno	Normalna	Ne	Da
Kišovito	Blaga	Normalna	Ne	Da
Sunčano	Blaga	Normalna	Da	Da
Oblačno	Blaga	Visoka	Da	Da
Oblačno	Topla	Normalna	Ne	Da
Kišovito	Blaga	Visoka	Da	Ne

Učenje (generalizacija) kao **pretraživanje**

Jedan skup pravila (**koncept**) koji ispravno klasificira primjere ako se primjenjuje po redoslijedu (kao lista) i ne nužno najbolji je:

1. *Ako vrijeme sunčano i vlažnost visoka, Tada igraj*
2. *Ako vrijeme kišovito i vjetrovito, Tada ne igraj*
3. *Ako vrijeme oblačno Tada igraj*
4. *Ako vlažnost normalna Tada igraj*
5. *Ako ništa od gornjega Tada igraj*

- Strojno učenje kao generalizacija (za razliku od statistike), može se predstaviti kao problem **pretraživanja velikog prostora raznih koncepata** za onim konceptom (rezultatom učenja) koji najbolje opisuje dane primjere (t.j. iskustvo).
- Ovdje su koncepti **svi skupovi pravila koji se mogu izgraditi temeljem 14 primjera i njihovih vrijednosti atributa.**
- Pojedini koncept (skup pravila) ne sadrži više pravila nego primjera (nije potrebno više od jednog pravila za svaki primjer). U našem primjeru svaki koncept ima od 1 do 14 pravila.

Učenje (generalizacija) kao **pretraživanje**

- Izbrojimo li sve atribute i pridružene vrijednosti (+ ciljni) , postoji 4 x 4 x 3 x 3 x 2 = 288 mogućnosti za svako pravilo (dodatna vrijednost atributa: NE sudjeluje u pravilu).
- Za koncept s do 14 pravila ukupno postoji 2,7 x 10^{34} koncepata.
- To je vrlo veliki prostor pretraživanja koncepata ali je **konačan.**
- Kada bi pojedini atributi imali numeričke vrijednosti prostor koncepata bio bi **beskonača**n. To se izbjegava **diskretizacijom**.
- Pretraživanje skupa se izvodi na dva načina: a)**uz eliminaciju** ili b) **penjanjem** (engl. hill climbing).
- Iscrpno pretraživanje skupa velikog broja koncepata je računalno nepraktično te se uvode brojni **heuristički postupci** (koji ne garantiraju pronalaženje optimalnog koncepta).

Pretraživanje uz eliminaciju

- Proces generalizacije (učenja) može se promatrati kao pretraživanje u ogromnom prostoru koncepata i **eliminaciju onih koncepata koji ne pokrivaju dane iskustvene primjere.**
- Sa svakim primjerom prostor koncepata se smanjuje. Pozitivan primjer eliminira sve koncepte koji ga ne opisuju. Negativan primjer eliminira sve koncepte koji ga opisuju.

Mogući završetak:

Svi primjeri su iskorišteni

- Ostao jedan koncept – to je **ciljni koncept** (rijetko se dogodi).
- Ostalo više koncepata – mogu se uporabiti za klasifikaciju uz:
  - Ako nepoznati primjer odgovara svim konceptima to je ciljna klasifikacija.
  - Ako nepoznati primjer odgovara nekim konceptima (ali ne svima) – postoji višeznačnost koja se mora riješiti **dodatnim kriterijima** za selekciju jednog "najboljeg" koncepta.

Svi primjeri nisu iskorišteni a svi koncepti su eliminirani

U podacima ima šuma (pogrešaka) ili je jezik za predstavljanje koncepata nedovoljno izražajan (ciljni koncept neuhvatljiv).

Pretraživanje uz penjanje

- Proces generalizacije (učenja) može se promatrati na drugi način ne kao pretraživanje skupova koncepata i eliminiranje već kao **postupak penjanja** (engl. *hill-climbing*) **u prostoru koncepata** u potrazi za opisom koji najbolje opisuje primjere.
- Obično se **započinje s najopćenitijim konceptom koji se pomalo specijalizira** (problem: koliku specijalizaciju dopustiti obzirom na preveliku prilagodbu podacima za učenje – engl. *overfitting*).
- **Tako radi većina postupaka strojnog učenja.**

Npr.: Da li bolje rezultate **na skupu za testiranje** daje pravilo a) ili b) ?

a) *Ako postoji (astigmatizam) i (normalno stvaranje suza), preporuka je tvrde kontaktne leće*

b) Ako postoji (astigmatizam) i (normalno stvaranje suza) i (minus dioptrija), preporuka je tvrde kontaktne leće

Pristranost u učenju (generalizaciji)

Proces generalizacije (učenja) traži **donošenje odluka** o:

- Određivanje jezika za opis koncepata (pristranost jezika).
- Način pretraživanja prostora koncepata (pristranost pretraživanja).
- Postupak izbjegavanje slaganja s podacima za učenje (eng. *overfitting*).

Pristranost jezika (engl. *language bias*)

- Potrebno je pronaći i uporabiti "univerzalan" jezik koji omogućuje **izražavanje svih mogućih koncepata** (opisa podskupova primjera).
- Svi podskupovi primjera se mogu opisati ako jezik koncepta dozvoljava logičku disjunkciju (ILI) među izjavama. Ako je koncept skup pravila, disjunkcija se realizira odvojenim pravilima.
- Formalan jezik **predikatne logike** je vrlo izražajan, ali je proces indukcije vrlo složen.
- Ako postoji više opisa jednog koncepta, najbolji je najjednostavniji opis.

  Ockhamova oštrica (minimizacija pretpostavki): "Ako imate dvije teorije koje opisuju isto, odaberite jednostavniju."

  Primjer: Kroz 2 točke moguće beskonačno krivulja – najbolje pravac.

Pristranost u učenju (generalizaciji)

Pristranost u načinu pretraživanja prostora koncepata (engl. *search bias*)

- Računalno nije izvedivo pretraživanje cijelog prostora koncepata.
- Posljedica toga je nemogućnost garancije optimalnog rješenja.
- Primjerice pohlepni algoritmi (engl. *greedy*) traže jedino po jedno pravilo i dodaju ga skupu pravila (konceptu). Moguće je međutim da par (2) pravila uspješnije pokrivaju primjere nego dva pojedinačno.
- Mnogi postupci odabiru na početku jedan atribut i dalje dodaju druge. Bolje je ne obvezati se na početni fiksni atribut (ma kako dobro izdvojen), već stalno imati nekoliko aktivnih alternativa (engl. *beam search*).
- Većina postupaka strojnog učenja (iako ne svi) provodi **općenitiji prema specifičnom** pristupu (eng. *general-to-specific*); t.j. počinje s jednim (pojedinačno „najboljem") atributom pa dodaje druge.

Pristranost u učenju (generalizaciji)

Pristranost u izbjegavanju slaganja s podacima za učenje
(engl. *overfitting bias*)

- Prevelika specijalizacije neće dobro opisivati nove (dotad neviđene) primjere – koncept se previše prilagodio podacima za učenje.

- Postupci izbjegavanje prevelikog slaganja s podacima za učenje:

  - Koncept se **ne specijalizira do kraja** (engl. *forward pruning*). Npr. pratimo uspješnost klasifikacije ili uporabivost (engl. *actionable*) koncepta.
  - Koncept se **specijalizira do kraja ali se zatim poopćava unatrag** (emgl. *backward pruning*). Mjerimo uspješnost.
  - Uključenje **što više slučajnosti u postupak učenja** (unakrsna međuvalidacija, slučajne šume i sl.).

Tipični primjeri zadataka otkrivanja znanja

Otkrivanje znanja- primjeri zadataka

Vrlo jednostavan i **nerealan** primjer:

Temeljem prošlih igara (**E** - **iskustvo**) odredi da li igrati ili ne tenis (**T** - **zadatak**) za slučaj koji nije naveden u tablici.

Atributi (značajke) i njihove vrijednosti — Ciljni atribut

Prošle igre, primjeri, **iskustvo**

(enl. Experience)

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne
Oblačno	Hladno	Normalna	Da	Da
Sunčano	Blaga	Visoka	Ne	Ne
Sunčano	Hladno	Normalna	Ne	Da
Kišovito	Blaga	Normalna	Ne	Da
Sunčano	Blaga	Normalna	Da	Da
Oblačno	Blaga	Visoka	Da	Da
Oblačno	Topla	Normalna	Ne	Da
Kišovito	Blaga	Visoka	Da	Ne

Otkrivanje znanja- primjeri zadataka

Temeljem skupa zapisa EKG-a (iskustva) **klasificirati** novi EKG kao poremećaj ili b.o. , te probati p**redvidjeti** poremećaj (dva zadatka).

- PAF (paroxysmal atrial fibrilation) baza podataka sadrži **100** parova 30-minutnih zapisa EKG.a.
- Svaki par je izvučen iz 24-satnih zapisa.
- Grupa A ima poremećaj PAF (bolesni). Za svakog pacijenta jedan EKG zapis izvučen je neposredno prije PAF-a, a drugi zapis vremenski udaljen od PAF-a.
- Grupa N nema poremećaj PAF (zdravi). EKG je izvučen slučajno iz 24-satnog zapisa.
- Cijeli skup EKG-a je podijeljen na podskup za učenje i podskup za testiranje.

Otkrivanje znanja- primjeri zadataka

Neki primjeri analize podataka u domeni znanosti o životu (engl. Life sciences)

- Molekularno strojno učenje (otkrivanje i oblikovanje novih molekula željenih svojstava).
- Biofizičko strojno učenje (interakcija molekule lijeka s proteinima).
- Genetika (istraživanje pojedinih gena i uloge) i genomika (kolektivna karakterizacija gena).
- Primjena u mikroskopiji (klasifikacija ii segmentacija stanica u mikroskopskoj slici).
- Medicina (dijagnostika, predikcija, modeliranje problema). Analiza Electronic Health Record (EHR) skupova podataka. Problem formatiranja i standardiziranja podataka iz različitih repozitorija (a.k.a. Biobanks). Npr. https://www.ukbiobank.ac.uk/
- Radiologija (Medicine Image Analysis). Slike iz različitih izvora. Zadaci: klasifikacija, segmentacija, opisi.

Otkrivanje znanja- primjeri zadataka

Primjer iz financijskog područja osiguranja (međunarodno natjecanje COIL, tvrtka iz Nizozemske, (EU Network of Excellence)

Baza klijenata:

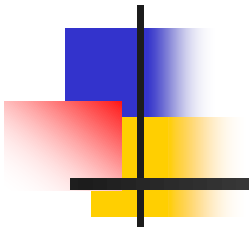
- Skup za učenje: 5822 klijenata je osiguralo auto, a 348 (od 5822) je osiguralo i auto prikolicu (poznato).
- Skup za testiranje: 4000 klijenata je osiguralo auto. 238 je osiguralo i auto prikolicu (koji su, to zna samo organizator).
- Svaki klijent je opisan s 43 demografskih značajki (primanja, gdje stanuje, obiteljski status, …), te s 42 poslovnih značajki (broj polica, premija, …). Ukupno 85 značajki.
- Zadatak 1: **Pronađi** 20% vlasnika u testnom skupu (0.2 x 4000 = 800) tako da taj podskup od 800 sadrži što više vlasnika dodatne police (za prikolicu). Idealno svih 238.
- Zadatak 2: **Opiši** te ciljane klijente (odredi im profil).

Otkrivanje znanja- primjeri zadataka

Neki primjeri analize podataka u domeni upravljanja odnosa s klijentima

Engl. **Customer relationship management** (**CRM**)

- Proces istraživanja interakcije poslovne organizacije s klijentima zasnovan na dubinskoj analizi velikog broja podataka u ali i izvan organizacije (vlastito web sjedište, telefonski razgovori, e-pošta, promotivni materijali te socijalne mreže.
- Istraživanje osigurava poslovnoj organizaciji saznanja o ciljanoj skupini klijenata, njihovim potrebama i ponašanju, sve kako bi se osigurao rast prodaje roba i usluga.
- CRM se koristi za prošle, sadašnje i buduće klijente.
- Proces zaključno obuhvaća specifičan pristup ciljanoj skupini klijenata.
- CRM koncept preko upitnika razvijen je još 1970. god.



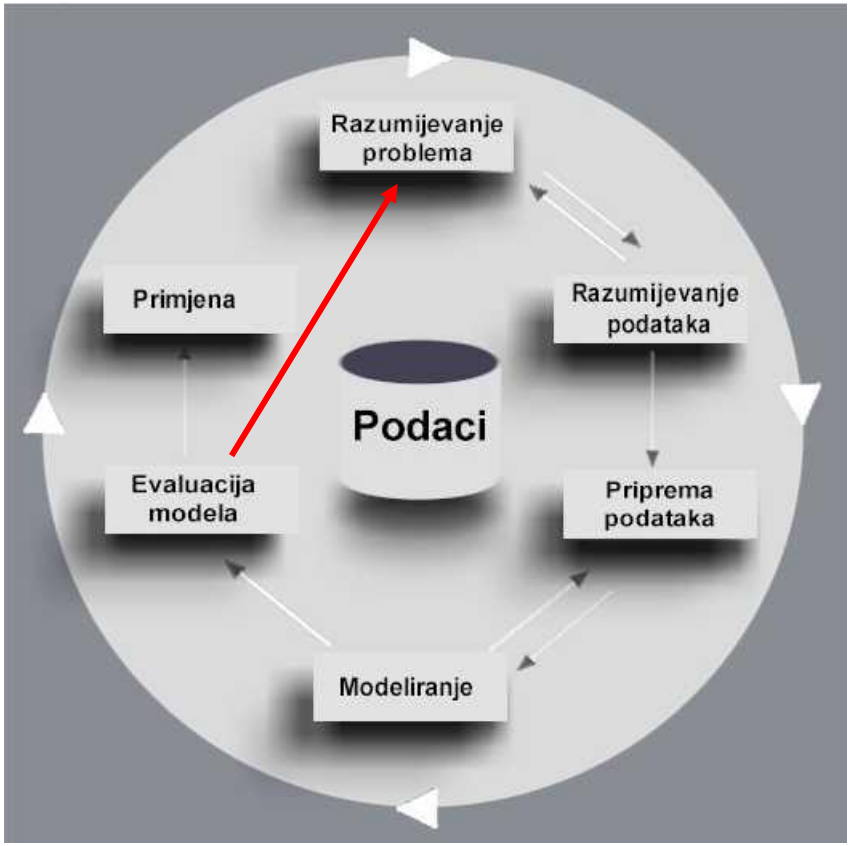
Otkrivanje znanja u skupovima podataka

- Uvodna razmatranja
- Proces dubinske analize
- Ulazni podaci
- Oblici induciranog znanja

Proces dubinske analize podataka

Proces dubinske analize podataka

Proces dubinske analize podataka je visoko itarativan i oslanja se na kontinuirano eksperimentiranje uz promjene parametara u pojedinim fazama procesa.



Proces dubinske analize podataka

Razumijevanje problema i podataka:
80% važno, 20% vremena.
Priprema podataka, modeliranje, evaluacija:
20% važno, 80% vremena.

Razumijevanje problema

- Ciljevi u domeni (npr. kancerogenost određenog kem. spoja).
- Kriteriji uspješnosti projekta (npr. otkrivanje potencijalno opasnih tvari).
- Posebni ciljevi analize podataka (npr. klasifikacijski model visoke točnosti).
- Plan izvedbe.

Razumijevanje podataka

- Prikupljanje početnih podataka (lokacija).
- Opis podataka (volumen, značenje varijabli/atributa/značajki).
- Istraživanje podataka (distribucija vrijednosti- **stratifikacija**).
- Verifikacija kvalitete podataka (neodređene vrijednosti, pogreške, "*outliers*").

Proces dubinske analize podataka

Priprema podataka (ETL – extract, transform, load - Talend)

- Selekcija podataka.
- Čišćenje podataka (normalizacija, ograničenja na vrijednosti).
- Konstruiranje novih podataka (nedostajuće vrijednosti).
- Formatiranje podataka (ovisno o alatu i postupku).

Priprema podataka uporabom filtara u alatu (npr. WEKA)

- ***AttributeExpressionFilter*** – stvara novu značajku/atribut podataka uporabom nekih matematičkih operacija na postojećim značajkama.
- ***DiscretizeFilter*** – diskretizira raspon kontinuiranih numeričkih vrijednosti značajki u nominalne (ne-numeričke) vrijednosti.
- ***InstanceFilter*** – izbacuje primjere iz skupa koji imaju određenu vrijednost nominalnih značajki ili raspon numeričkih.
- ***MakeindicatorFilter*** – kreira novu značajki s binarnim vrijednostima 0 i 1 prema određenim rasponima numeričkih i nominalnih vrijednosti.
- ***ObfuscateFilter*** – radi zaštite podataka značajki/atributu mijenja naziv i ndeksirane simboličke vrijednosti (A1, A2, …), a nominalnim vrijednostima mijenja u drugi skup indeksiranih vrijednosti (V1, V2, …).
- … (oko 50 filtara)

Proces dubinske analize podataka

Modeliranje

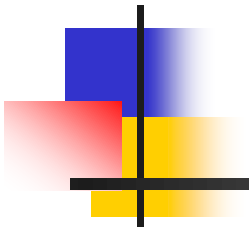
- Izbor tehnike modeliranja (klasifikacija, predikcija, opis skupine, segmentacija, …). **Ovisno o cilju analize (konceptu).**
- Oblikovanje ispitivanja modela (definiranje metode testiranja).
- Izgradnja modela (pravila, klasifikacijsko stablo, …).
- Validacija modela (prema kriterijima dubinske analize) – pokrivanje ciljeva.

Evaluacija modela (iz perspektive domene)

- Evaluacija rezultata (neočekivano i uporabivo ?)
- Evaluacija procesa (kontrola kvalitete)
- Određivanje slijedećeg koraka

Postavljanje modela u korisničko okruženje

- Nadzor i održavanje modela
- Završno izvješće (podrška odlučivanju, "on-line" algoritam, …)



Otkrivanje znanja u skupovima podataka

- Uvodna razmatranja
- Proces dubinske analize
- Ulazni podaci
- Oblici induciranog znanja

Ulazni podaci

Dubinska analiza podataka – ulazni podaci

- Iskustvo u dubinskoj analizi podataka predstavljeno je **primjerima** ili **instancijama**.
- Svaki primjer predstavljen je skupom **značajki** ili **atributa**.
- Značajke ili atributi mogu imati **numeričke** ili **nominalne** (nenumeričke vrijednosti)

značajke / atributi

ciljni atribut

Primjeri

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne

Vrijednosti atributa (ovdje samo nominalne)

Dubinska analiza podataka – ulazni podaci

Primjer ulaznih podataka opisan **tablicom** (iako najčešće upotrebljavan) vrlo je restriktivan opis neke domene.

Primjer: Želimo inducirati koncept "sestra".

A1	A2	Sestra A2 od A1
Osoba_x1	Osoba_x2	DA
Osoba_x3	Osoba_x4	NE
...	...	...

Iz gornje tablice nije moguće poopćiti koncept (testira se samo vrijednost).

Relacijska tablica:

Ime_A	Spol_A	Rdtlj_1A	Rdtlj_2A	Ime_B	Spol_B	Rdtlj_1B	Rdtlj_2B	B= Sestra
...								DA
							...	NE

Teže, ali moguće je inducirati pravilo za "sestra":

AKO (Spol_B = žensko) i ((Rdtlj_1A = Rdtlj_1B)ili (Rdtlj_2A = Rdtlj_2B))

Dubinska analiza podataka – ulazni podaci

Generiranje tablice podataka koja opisuje domenu problema (dio *PRIPREMA PODATAKA* u procesu dubinske analize) posebno je područje istraživanja: "**Skladištenje podataka**" (engl. Data warehousing):

A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.

Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Dubinska analiza podataka – ulazni podaci

Vrijednosti značajki/atributa

- **Numeričke** vrijednosti (cijeli brojevi, realni, razlomci, …).
- Numeričke **intervalne** vrijednosti – mjerene u jednakim jedinicama (npr. godine, iako neke matematičke operacije – množenje, nemaju smisla.
- Numeričke vrijednosti nad kojima su dozvoljene sve operacije nazivaju se **omjeri**.
- **Nominalne** (nenumeričke vrijednosti) – simboličke vrijednosti.
- Atributi koji imaju nominalne vrijednosti nazivaju se **kategorički**.
- Neke nominalne vrijednosti mogu se staviti u odnos poput numeričkih (npr. vruće – mlako – hladno).
- Vrijednosti atributa koje se mogu staviti u odnos nazivaju se **ordinalne** vrijednosti.
- Većina sustava prihvaća numeričke i nominalne vrijednosti atributa.

Dubinska analiza podataka – ulazni podaci

Najčešći problemi s ulaznim podacima

- **Nedostajuće vrijednosti**
  - Za nominalne vrijednosti atributa upisuje se nova vrijednost (npr. "NEDOSTAJE").
  - Za numeričke vrijednosti atributa upisuje se srednja vrijednost svih upisanih (podložno kritikama).
- **Nepripadajuće vrijednosti (engl. "outliers")**
  - Izbacujemo (potrebno ekspertno znanje domene).
- **Netočne vrijednosti**
  - Npr. pogrešno izmjerene ili unesene.
  - Ispravljamo ili izbacujemo (potrebno ekspertno znanje domene).
- **Dupliciranje primjera s istim vrijednostima svih atributa**
  - Izbacujemo duplikate primjera.

Zaključak:

Potrebno je dobro poznavanje domene i podataka s kojima radimo.

Priprema podataka (80% vremena)

Zašto je potrebna **ETL** priprema? – zbog **kvalitete podataka**

Više dimenzijski pogled, podaci ispravni ili ne, kompletni, konzistentni, vjerodostojni, vremenski ispravno prikupljeni, interpretabilni

Čišćenje podataka

Upis nedostajućih vrijednosti, glađenje podataka s šumom, identifikacija i izbacivanje nepripadajućih podataka (emgl. Outliers), razriješiti nekonzistencije (npr. *Age*="42", *Birthday*="03/07/2010")

Integracija podataka iz više izvora (baza podataka), Npr. Bill=William

Redukcija podataka

Dimenzijska (izbacivanje nevažnih atributa), brojčana (podatke nadomjestiti modelom), komprimiranje (bez i sa gubitkom, npr. MP-3)

Transformacija podataka

Normalizacija na odabrani raspon vrijednosti, hijerarhijsko predstavljanje atributa (npr. broj godina -> mlad, odrastao, senior)

Dubinska analiza podataka – ulazni podaci

Formatiranje ulaznih podataka – ovisno o alatu. **WEKA - arff**

```
% ARFF file for Breast cancer data
@relation breast-cancer
@attribute age {'10-19','20-29','30-39','40-49','50-59','60-69','70-79','80-89','90-99'}
@attribute menopause {'lt40','ge40','premeno'}
@attribute tumor-size {'0-4','5-9','10-14','15-19','20-24','25-29','30-34','35-39','40-
        44','45-49','50-54','55-59'}
. . .

@data
'40-49','premeno','15-19','0-2','yes','3','right','left_up','no','recurrence-events'
'50-59','ge40','15-19','0-2','no','1','right','central','no','no-recurrence-events'
'50-59','ge40','35-39','0-2','no','2','left','left_low','no','recurrence-events'
'40-49','premeno','25-29','0-2',?,'2','left','right_low','yes','no-recurrence-events'
. . .
```

Otkrivanje znanja u skupovima podataka

- Uvodna razmatranja
- Proces dubinske analize
- Ulazni podaci
- Oblici induciranog znanja

Najčešći oblici induciranog znanja

(predstavljanje strukturnih uzoraka u skupovima podataka)

Oblici induciranog znanja

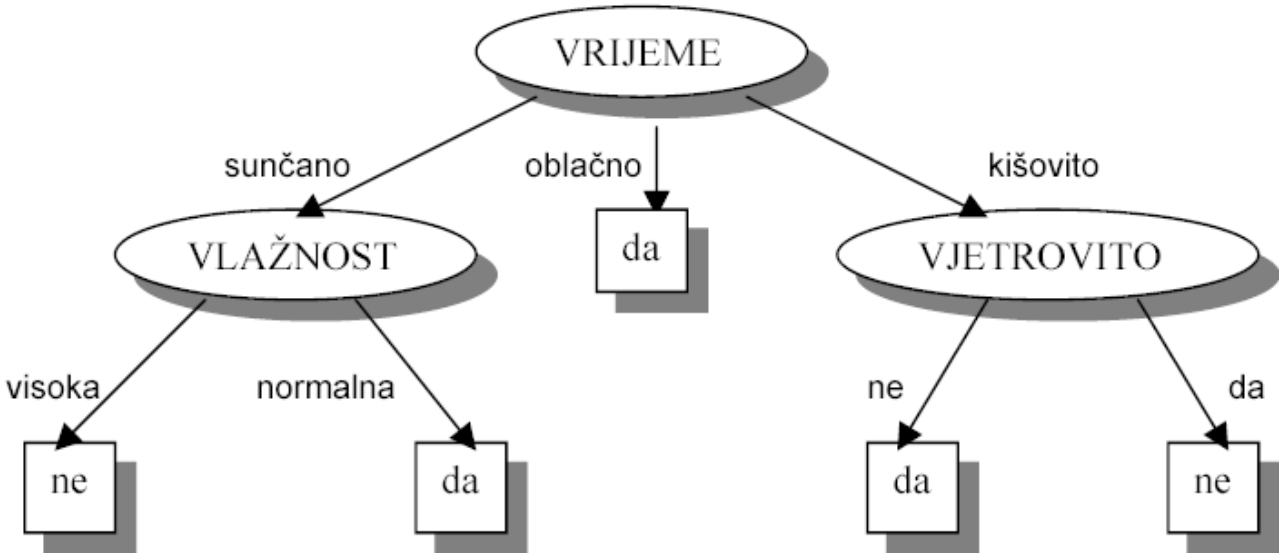
Najčešći oblici induciranog znanja

1. Stabla odlučivanja
2. Klasifikacijska pravila
3. Pravila pridruživanja
4. Predstavljanje znanja pojedinačnim primjerima
5. Predstavljanje skupina
6. Probabilistički postupci otkrivanja i predstavljanja znanja
7. Hibridni postupci otkrivanja i predstavljanja znanja
8. Ansambli klasifikatora
9. Funkcijski postupci otkrivanja i predstavljanja znanja

Oblici induciranog znanja

1. Stabla odlučivanja za klasifikaciju primjera

- Najjednostavniji i elementaran način predstavljanja znanja.
- Grafički prikaz stabla u kojem su atributi čvorovi a lukovi njihove moguće nominalne vrijednosti ili intervali numeričkih vrijednosti.
- Nepoznati primjer se vrijednosno uspoređuje po pojedinim čvorovima te krajnji čvorovi (lišće) daju konačnu klasifikaciju.
- **Primjer odlučivanjao igranju tenisa:**



Oblici induciranog znanja

Stabla odlučivanja

- Klasifikacijski i regresijski modeli
- Podijeli-pa-vladaj strategija izgradnje stabla (engl. *divide-and-conquer)*
- Izbor čvorova temeljen na informacijskoj vrijednosti.
- Poznatiji algoritmi ili porodice algoritama
  - ID3 (engl. *Induction of Decision Trees*),
  - CART (engl. *Classification and Regression Trees*),
  - ASSISTANT
  - C4.5
  - M5
  - ADTree (engl. *Alternating Decision Tree*)
  - Best-first tree
  - Funkcionalna stabla (engl. *Functional trees*)
  - Slučajna šuma (engl. *Random forest*)
  - . . .

Oblici induciranog znanja

2. Klasifikacijska pravila (kao koncept)

- Popularna alternativa stablima odlučivanja. Svako pojedinačno pravilo je grumen (engl. *nugget*) znanja.
- Oblik pravila:

  AKO (uvjet = konjunkcija testova) TADA (razred)
- U nekim pravilima uvjet mogu činiti različiti logički izrazi.
- Najčešće su pravila u konceptu povezana disjunkcijom (ILI). Ako bilo koje pravilo odgovara primjeru klasificira se prema TADA strani.
- Pravila i stabla odlučivanja su povezani.
  - Svaki put u stablu do završnog čvora je jedno pravilo. Preslikavanje stabla u skup pravila je jednostavno. Dobivena pravila su nedvosmislena, nije bitan redoslijed primjene pravila, ali su redundantna i nepotrebno složena.
  - Preslikavanje skupa pravila u stablo je složeno (zbog disjunkcije među pravilima koju nije jednostavno preslikati u stabla).

Oblici induciranog znanja

Klasifikacijska pravila (kao koncept)

Problemi s pravilima (ako nema dodatne informacije o uporabi):

Te vrste problema ne pojavljuju se u stablima odlučivanja.

- Redoslijed uporabe pravila može dovesti do različite klasifikacije. Konflikt se rješava dodatnim postupcima na više načina (npr. primjer se klasificira temeljem najčešće uporabljenog pravila).
- Primjer se ne klasificira (niti jedno pravilo ga ne opisuje). Jedno rješenje: klasifikacija u najčešći razred.
- Nezavisnost primjene klasifikacijskih pravila moguć je u **binarnoj klasifikaciji** (postoje 2 ciljna razreda i pravila opisuju jedan razred).
  - Pretpostavka zatvorenog svijeta: ako primjer nije u razredu 1, tada je u razredu 2.
  - Takva pravila predstavljena su u disjunkcijskom normalnom obliku (DNF):

    AKO [(... $\wedge$ ... $\wedge$ ...) $\vee$ (... $\wedge$ ... $\wedge$ ...) $\vee$ (... $\wedge$... $\wedge$ ...) ...], TADA ...
- Većina klasifikacijskih algoritama generira skup pravila gdje je redoslijed primjene bitan (zavisna pravila).

Oblici induciranog znanja

Klasifikacijska pravila s iznimkama

- Često novi primjeri uvode iznimke koje proširuju prikaz pravila.
- Skup pravila (koncept) se proširuje inkrementalno.

If petal-length $\geq$ 2.45 and petal-length $<$ 4.45 then Iris-versicolor

EXCEPT if petal-length $<$ 1.0 then Iris-setosa.

- Iznimke mogu biti ugnježđene EXCEPT EXCEPT …
- U razumijevanju takvih pravila fokusiramo se na pojedino pravilo (lokalni fokus) a ne na cijeli skup. Iznimka se odnosi samo na jedno pravilo.
- Formulacija EXCEPT je više psihološka neko formalno logička (If-Then-Else). Iznimke su rijetke a raniji testovi mnogo širi.

Oblici induciranog znanja

Klasifikacijska pravila - zaključak

- Sličan pristup kao kod stabala odlučivanja: odvoji-pa-vladaj strategija (engl. *seperate-and-conquer*) s razlikom:
  - Stabla: maksimizacija informacijskog dobitka dijeljenjem na nekom atributu
  - Pravila: odabir para atribut-vrijednost koji uzrokuje maksimizaciju vjerojatnosti točne klasifikacije

- Poznatiji algoritmi:
  - 1-R (engl. *OneRule*)
  - M5Rules
  - PRISM
  - PART
  - RIDOR (engl. *Ripple Down Rule Learner*)
  - Tablica odlučivanja (engl. *decision table*), prvog i drugog reda
  - RIPPER (engl. *Repeated Incremental Pruning to Produce Error Reduction*)
  - ...

Oblici induciranog znanja

3. Pravila pridruživanja (asocijacijska parvila) **– 1/2**

Slično kao ranija klasifikacijska pravila uz razliku da mogu predviđati **bilo koji atribut** a ne samo ciljni (razred).

Npr. za igranje tenisa:

AKO (*Temperatura=Hladno*) TADA (*Vlažnost=Normalna*)

- Iz skupa podataka može se izvesti veliki broj pravila pridruživanja koja pokazuju različite regularnosti u podacima.
- Zbog velikog broja pravila pridruživanja fokus je na onima koja **pokrivaju** veliki podskup primjera i u tom podskupu imaju visoku **preciznost**.

Mjere za pravila pridruživanja:

Pokrivanje (engl. *coverage*) ili **potpora** (engl. *support*) = broj primjera koje pravilo ispravno predviđa (često u u odnosu na cijeli skup).

Preciznost (engl. *accuracy*) ili **uvjerenost/pouzdanost** (engl. *confidence*) = broj primjera koje pravilo ispravno predviđa u odnosu na broj primjera na koje se odnosi.

Oblici induciranog znanja

Tablica igranja tenisa i pravila pridruživanja

Vrijeme	Temperatura	Vlažnost	Vjetrovito	Igrati
Sunčano	Topla	Visoka	Ne	Ne
Sunčano	Topla	Visoka	Da	Ne
Oblačno	Topla	Visoka	Ne	Da
Kišovito	Blaga	Visoka	Ne	Da
Kišovito	Hladno	Normalna	Ne	Da
Kišovito	Hladno	Normalna	Da	Ne
Oblačno	Hladno	Normalna	Da	Da
Sunčano	Blaga	Visoka	Ne	Ne
Sunčano	Hladno	Normalna	Ne	Da
Kišovito	Blaga	Normalna	Ne	Da
Sunčano	Blaga	Normalna	Da	Da
Oblačno	Blaga	Visoka	Da	Da
Oblačno	Topla	Normalna	Ne	Da
Kišovito	Blaga	Visoka	Da	Ne

Oblici induciranog znanja

Pravila pridruživanja (asocijacijska parvila) **– 2/2**

Pravilo: AKO (*Temperatura=Hladno*) TADA (*Vlažnost=Normalna*)

Iz tablice je evidentno:

Pravilo **pokriva** 4 primjera (katkad oznaka 4 od svih 14 ili 29%).
To su svi primjeri u tablici u kojima je
(*Temperatura=Hladno i Vlažnost=Normalna*).

Pravilo ima 100% **preciznost** jer svi hladni dani (AKO strana) imaju normalnu vlažnost (TADA strana). Ne postoji *Temperatura=Hladno* a da nije i *Vlažnost=Normalna*

Preciznost se može izraziti preko uvjetne vjerojatnosti uzoraka (primjera):
P(TADA | AKO) = P(TADA i AKO) / P(AKO) = 4 / 4 = 1

Uobičajeno je **specificirati minimalno pokrivanje i preciznost** te potražiti sva pravila koja zadovoljavaju te kriterije.

Oblici induciranog znanja

4. Predstavljanje znanja pojedinačnim primjerima – 1/3

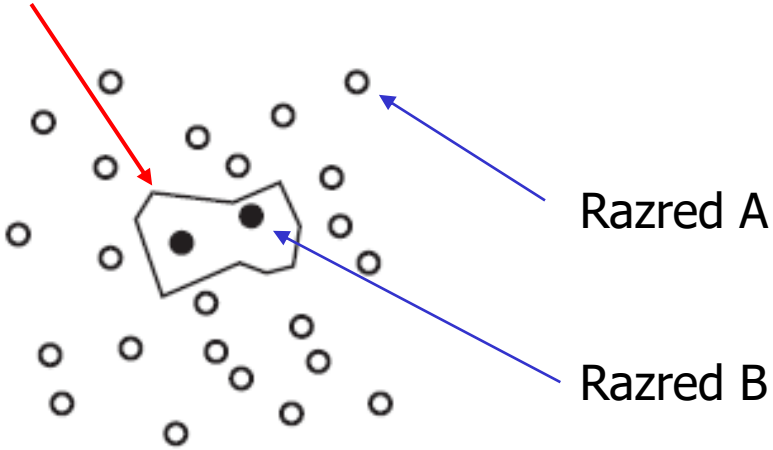
(engl. *instance based learning*)

- Najjednostavniji oblik učenja je **pamćenje pojedinačnih primjera zajedno s njihovom klasifikacijom** (pripadnost razredu).
- **Novi primjer** po dolasku u sustav traži sebi "najsličniji" prije memorizirani primjer i klasificira se u isti razred kao i taj "najsličniji".
- Umjesto izgradnje modela, primjer se klasificira tek po dolasku (sustav je "lijen").
- Komparacija s memoriranim primjerima i razvrstavanje prema "najsličnijem" naziva se klasifikacijska **metoda najbližih susjeda**.
- Komparacija udaljenosti je jednostavna ukoliko primjeri posjeduju samo numeričke značajke. To podrazumijeva normalizaciju vrijednosti značajki što može predstavljati problem.
- Kod nominalnih značajki vrijednosti se potpuno podudaraju ili ne, ali mogu se uključiti i složene metode (npr. vrijednost narančasto je bliže crvenom nego plavom).
- Neki atributi su značajniji od drugih te je potrebno uvesti težinske mjere.

Oblici induciranog znanja

Predstavljanje znanja pojedinačnim primjerima – 2/3

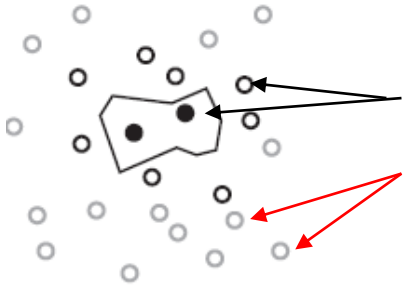
- Nije efikasno memorirati **sve** primjere. To usporava proces klasifikacije i zauzima nepotrebno veliki memorijski prostor.
- Neke vrijednosti atributa su vrlo malo promjenljive, pa je moguće pamtiti samo nekoliko reprezentativnih uzoraka iz središta regije uzoraka koji pripadaju istom razredu. Problem određivanja tih primjera.
- Pojedinačni primjeri ne izlučuju naučenu eksplicitnu strukturu ali zajedno s metrikom udaljenosti ipak opisuju neku regiju istog razreda predstavljenu poligonom.



Oblici induciranog znanja

Predstavljanje znanja pojedinačnim primjerima – 3/3

- Pri odbacivanju nerelevantnih primjera ostaju samo prototipovi razreda za izračunavanje "sličnosti". Prikaz pretpostavlja numeričke vrijednosti.

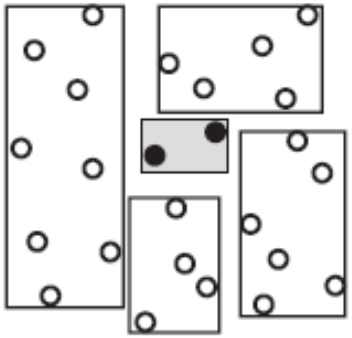


ostaju kao prototipovi dva razreda

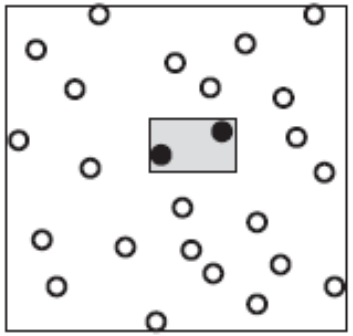
izbačeni

- Neki postupci predstavljanje znanja pojedinačnim primjerima mogu eksplicitno **generalizirati** primjere kreiranjem pravokutnih regija (a) koje mogu biti i ugniježđene (b). Problem kako definirati granice.

(a)



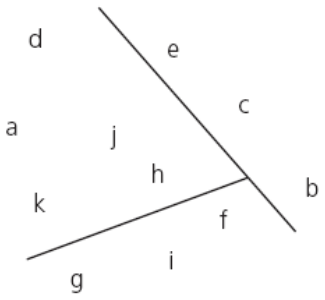
(b)



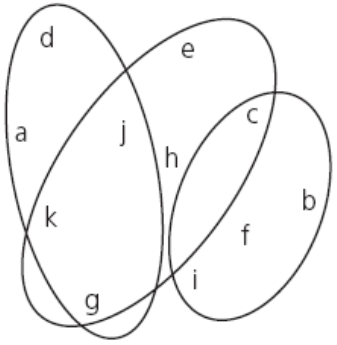
Oblici induciranog znanja

5. Predstavljanje skupina (grupa, klastera)

- Izlazno znanje predstavlja opis **kako pojedinačni primjeri tvore skupine**. To je **strukturni opis**, a u najjednostavnijem obliku predstavlja pridruživanje oznake skupine svakom primjeru (slika (a)).
- Neki postupci opisivanja skupina dozvoljavaju da primjer pripada više nego jednoj skupini (slika (b)),neki pak pridružuju primjer skupini po vjerojatnosti (slika (c)) dok neki izvode hijerarhijsku strukturu skupina - **dendogram** – stablasti dijagram (slika (d)).
- Predstavljanje skupina najčešće slijedi nakon indukcije stabla odlučivanja ili pravila koja alociraju primjer u skupinu



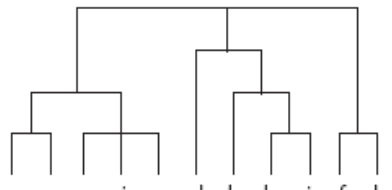
(a)



(b)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

(c)



(d)

Oblici induciranog znanja

6. Probabilistički postupci otkrivanja i predstavljanja znanja

- Cilj: što točnija procjena distribucije uvjetne vjerojatnosti za vrijednosti ciljnog atributa uz određene vrijednosti prediktorskih atributa.
- Bayesova mreža – usmjereni aciklički graf čiji su čvorovi atributi.
- Apriorne vjerojatnosti se zadaju ili dobivaju iz podataka.

- Poznatiji algoritmi:
  - Naivni Bayes (engl. *Naive Bayes*),
  - LBR (engl. *Lazy Bayesian Learning*),
  - SP-TAN (engl. *Super Parent Tree Augmented Naive Bayes*)
  - Bayesian multinets
  - AODE (engl. *Aggregating one-dependence estimators*)
  - …

Oblici induciranog znanja

7. Hibridni postupci otkrivanja i predstavljanja znanja

- Kombinacija dvaju ili više osnovnih klasifikacijsko-predikcijskih postupaka s ciljem dobivanja što učinkovitijeg modela.

- Primjeri
  - Stabla s logističkim modelom (engl. *Logistic model trees*)
  - Stabla s naivnim Bayesom u listovima (engl. *Naive Bayes trees*, NBTrees)
  - Tablice odlučivanja s naivnim Bayesom (engl. *Decision Trees Naive Bayes*, DTNB)
  - …

Oblici induciranog znanja

8. Ansambli klasifikatora

Imaju za cilj redukciju varijance i pristranosti ("overfitting") da bi dobili minimalnu pogrešku klasifikacije / predikcije

- **Sustavi više stručnjaka jednakih važnosti** (težina). Glasovanje za nominalni ciljni atribut ili srednja vrijednost za numerički. Isti skup podataka za učenje uz ponovljeni slučajan izbor (engl. resampling).
  - *Bagging, Voting, Grading, Stacking*
- **Višekaskadni sustavi**. Izlaz jednog modela je ulaz u drugi model (posebice primjeri koji se nisu dobro klasificirali ranije). Glasovanje, ali težina modela je proporcionalna njegovoj uspješnosti.
  - *Boosting (Adaboost), Multiboosting*
- Neki od najuspješnijih **ansambala** danas:
  - *Random forest, Immune network, Rotating forest*

Oblici induciranog znanja

9. Funkcijski postupci otkrivanja i predstavljanja znanja

- Opis odnosa ulaz-izlaz na temelju određene funkcije.
- U općenitom slučaju funkcija je nelinearna kombinacija prediktorskih (nezavisnih) varijabli, no zbog prirode odnosa u svijetu i interpretacije, često je dovoljno promatrati linearnu kombinaciju.

- Algoritmi i porodice algoritama:
  - Linearna regresija (engl. *linear regression*),
  - Logistička diskriminacija (regresija) (engl. *logistic discrimination*) – pripada i probabilističkim postupcima
  - Support-vector machines algoritmi (SVM)
  - Perceptron
  - Winnow (sličan Perceptronu)
  - Neuronske mreže (engl. *neural networks*)
  - Neuronske mreže za **duboko učenje** – **danas vrlo česte**
  - …

Deep Learning

Views on machine learning

View of machine learning in terms of three general approaches:

1. **Classical machine learning** techniques, which make predictions directly from a set of features that have been pre-specified by the user.
2. **Representation learning techniques**, which transform features into some intermediate representation prior to mapping them to final predictions (ex. PCA).
3. **Deep learning techniques**, a form of representation learning that uses **multiple transformation steps** to create very complex features

Slike preuzete iz. Weka, Data Mining:
Practical Machine Learning Tools and Techniques, 4th edition

Deep Learning

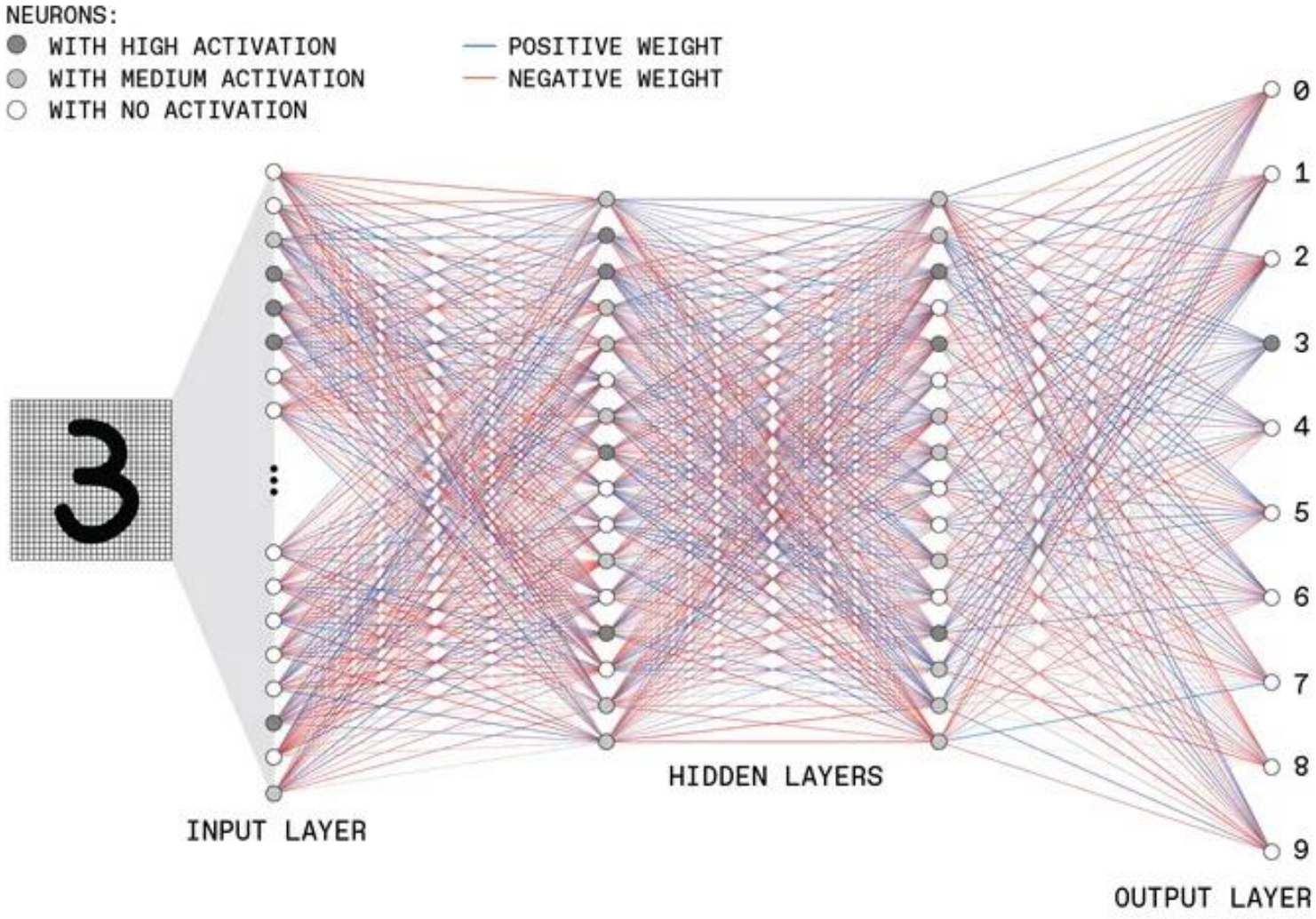
Deep Learning:

- Multiple levels of representation are obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level.

- The key aspect of deep learning is that these layers of features are not designed by human engineers: features are learned from data using a general-purpose learning procedure.

- Deep neural networks, are **artificial neural networks (ANN)** in which several layers of nodes are used to build up progressively more abstract representations of the data.

Deep Learning

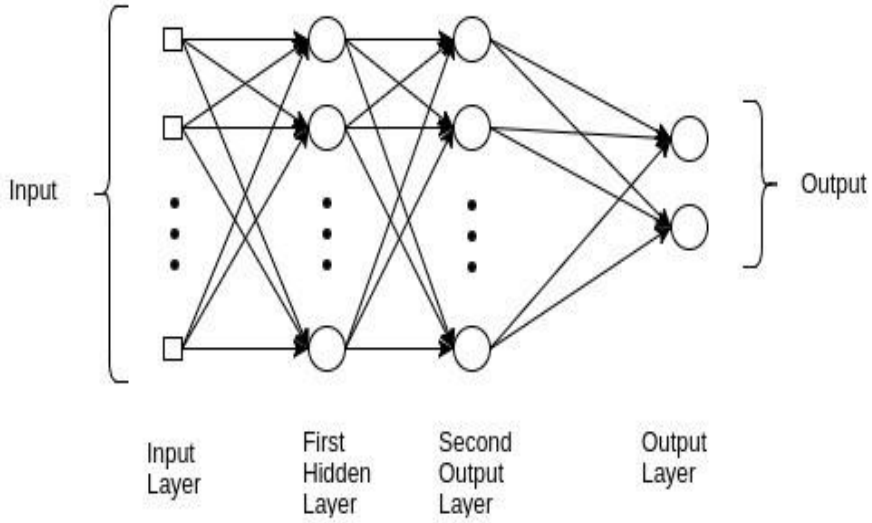


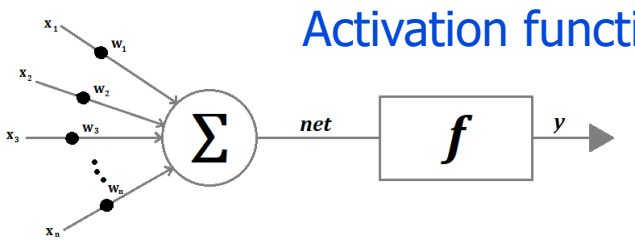
Deep Learning

Basic ANN: **Multilayer perceptron (MLP) – feed forward network**

Invented 1943, machine implementation 1958.

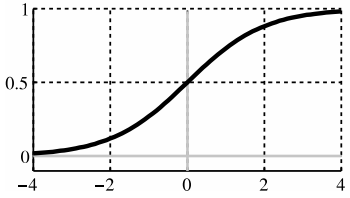
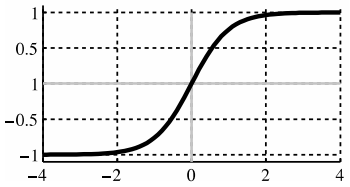
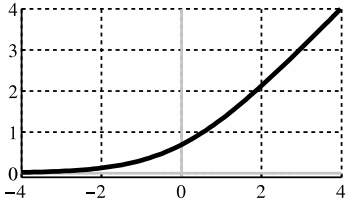
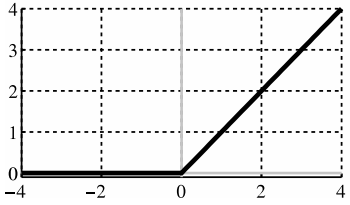
- The MLP consists of three or more layers (input , output layer with one or more hidden layers) of nonlinearly-activating nodes.
- Each node in one layer connects with a certain weight to every node in the following layer.
- It can distinguish data that is not linearly separable.





Deep Learning

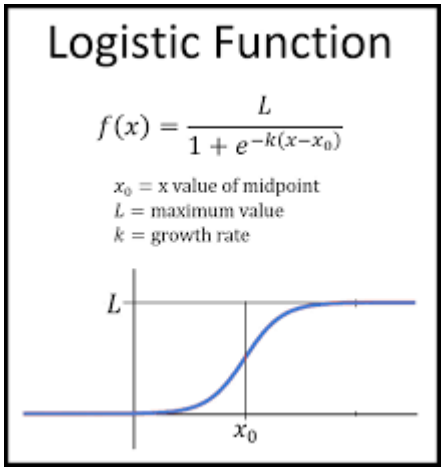
Non-linear activating function

Name and Graph	Function	Derivative
sigmoid(x) 	$h(x) = \frac{1}{1 + \exp(-x)}$	$h'(x) = h(x)[1 - h(x)]$
tanh(x) 	$h(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$	$h'(x) = 1 - h(x)^2$
softplus(x) 	$h(x) = \log(1 + \exp(x))$	$h'(x) = \frac{1}{1 + \exp(-x)}$
rectify(x) 	$h(x) = \max(0, x)$	$h'(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$

Deep Learning

Multilayer perceptron (MLP) – feed forward network

- For binary classification the last layer is usually the logistic function, and softmax (softargmax) for multi-class classification (normalizes outputs into [0, 1]), while for the hidden layers this was traditionally a sigmoid function, but today rectifier (ramp, ReLU) being common.
- During model training, the input-output pair is fixed, while the weights vary, and the network ends with the loss function.
- During model evaluation, the weights are fixed, while the inputs vary (and the target output may be unknown), and the network ends with the output layer (it does not include the loss function).



Deep Learning

Multilayer perceptron (MLP) – feed forward network

Learning perceptron by backpropagation (1974)

- The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight, iterating backwards one layer at a time from the last layer to the first.
- The loss function is usually logistic loss.
- With the chain rule of partial derivatives, we can represent gradient of the loss function as a product of gradients of all the activation functions of the nodes with respect to their weights. Therefore, the updated weights of nodes in the network depend on the gradients of the activation functions of each node.
- If the activation function is sigmoid, in deep network gradients product can lead to vanishing gradient (no change to the weight).
- By the early 2000s neural network methods had fallen out of favor again – kernel methods like SVMs yielded state of the art results on many problems.

Deep Learning

What influenced the **revival** of neural networks?

- The neural network „renaissance” (massive resurgence of interest in neural networks and deep learning techniques):
  - Starting around 2012, impressive results were achieved on long-standing problems in speech recognition and computer vision.
- Neural networks gets better with:
  - More data („Big data”)
  - Bigger models
  - More computation (Backpropagation in vector matrix form)
- Deep learning join many multilayer perceptrons together, so that there isn't just one hidden layer but many hidden layers. The "deeper" that the deep neural network is, the more sophisticated patterns the network can learn.
- **Brake through:** „Back propagation” calculation (learning) in vector – matrix form and computing with Grapgics Processing Units (**GPU**).

Examples of deep learning models

Convolutional neural networks (CNNs) - 1/3

- Application. Image processing (MLP would have too many weights)
- CNN has four major components:
  - Convolutional layers
  - Subsampling/pooling layers
  - Activation functions (e.g. rectifier = ReLU)
  - Fully connected layers
- The convolutional layers takes images as input and convolves (matrix product) filters (kernels) through sliding portions of an image obtaining feature map whicch is then processed by rectifier functions (ReLU).
- The filter is a small matrix, (e.g. 3x3) learned during training together with fully connected layers from initial random values.
- There can be several filters (hyper-parameter; correlated with number of attributes of an object), hence several feature maps.
- The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image.

Examples of deep learning models

Convolutional neural networks (CNNs) – 2/3

- Pooling layer is responsible for reducing the spatial size of the obtained Convolved Feature maps (e.g. witth 2x2 filter by average or max operation).
- Pooling filter is specified (not learned).
- The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network (many layers).
- Non-linear combinations of the high-level features as represented by the output of the convolutional layer.
- The activation functions control how the data flows from one layer to the next layer. ReLU sets minus values to 0.
- The last set of feature maps are flattened and given to the fully connected layer (similar to MLP) for classification.
- The parameters of the fully conneccted layer (together with filters), are learned by backpropagation during training

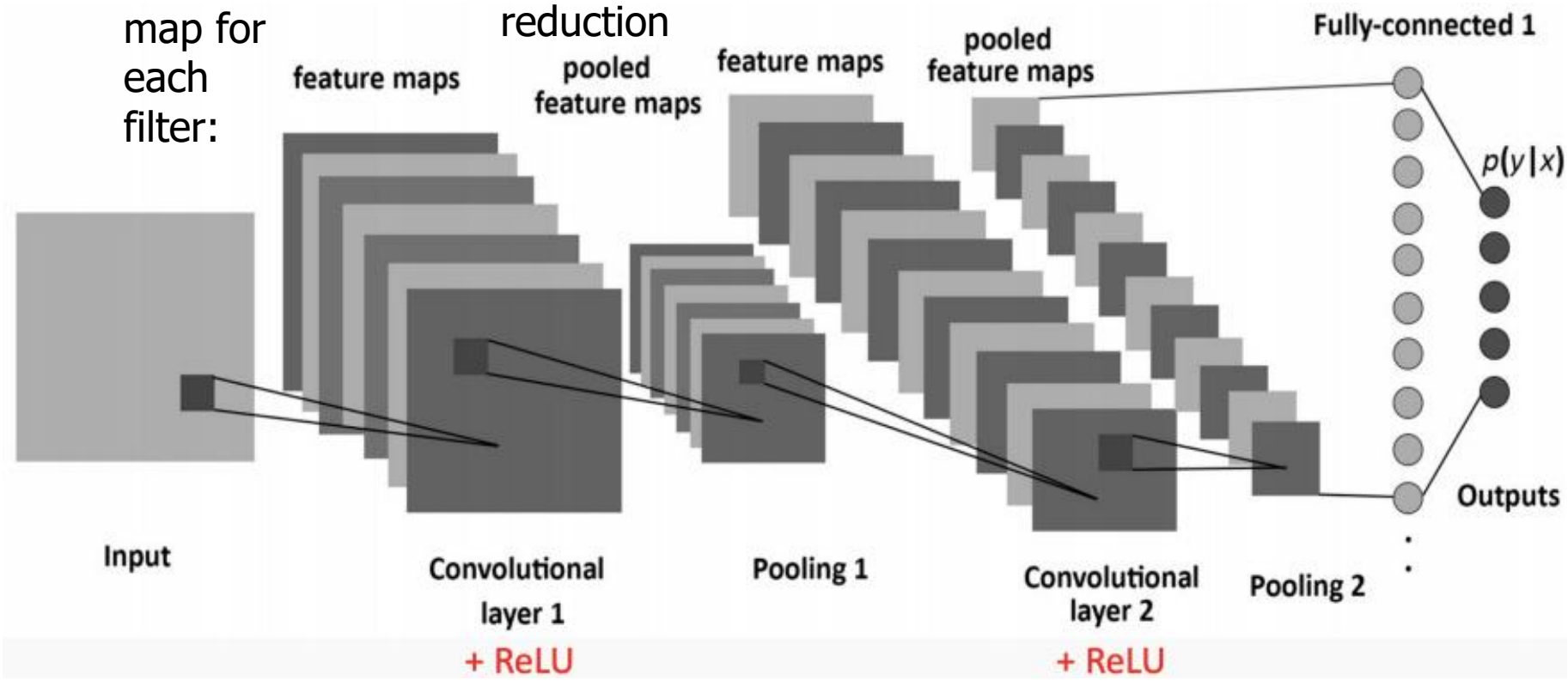
- Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition.

CNN Architecture

Different feature map for each filter:

Dimensionality reduction

Long 1-dim vector



Simple patterns
- less filters

Complex patterns
- more filters

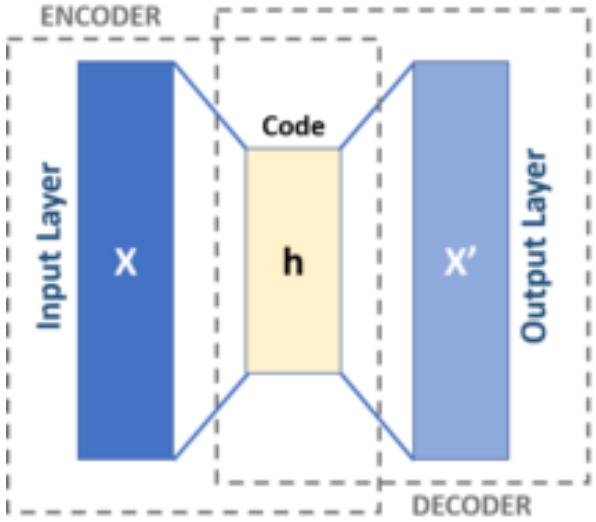
Izvor: M. Stewart, Simple Introduction to Convolutional Neural Networks, 2019.

Examples of deep learning models

Autoencoders

- Type of ANN used to learn efficient data codings in an unsupervised manner (better than PCA).
- The aim is to learn a representation (encoding) for a set of data, typically for dimensionality reduction or feature learning.
- Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the encoding a representation as close as possible to its original input.
- Having an input layer, an output layer and one or more hidden layers connecting them – where the output layer has the same number of nodes (neurons) as the input layer, and with the purpose of reconstructing its inputs (minimizing the difference between the input and the output) instead of predicting the target value given inputs.
- Training of an autoencoder is performed through Backpropagation of the error, just like a regular feedforward neural network.

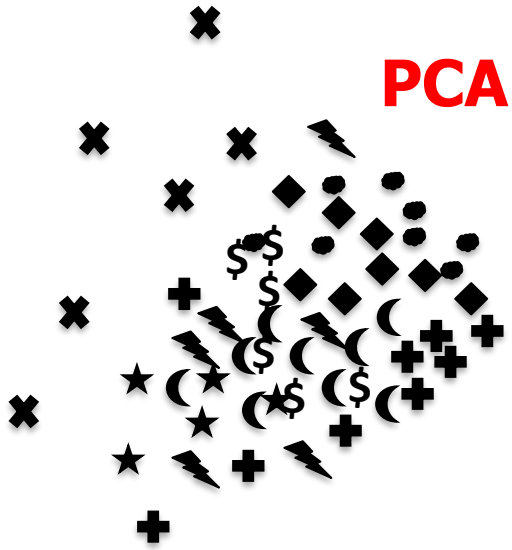
Autoencoder and comparison with PCA

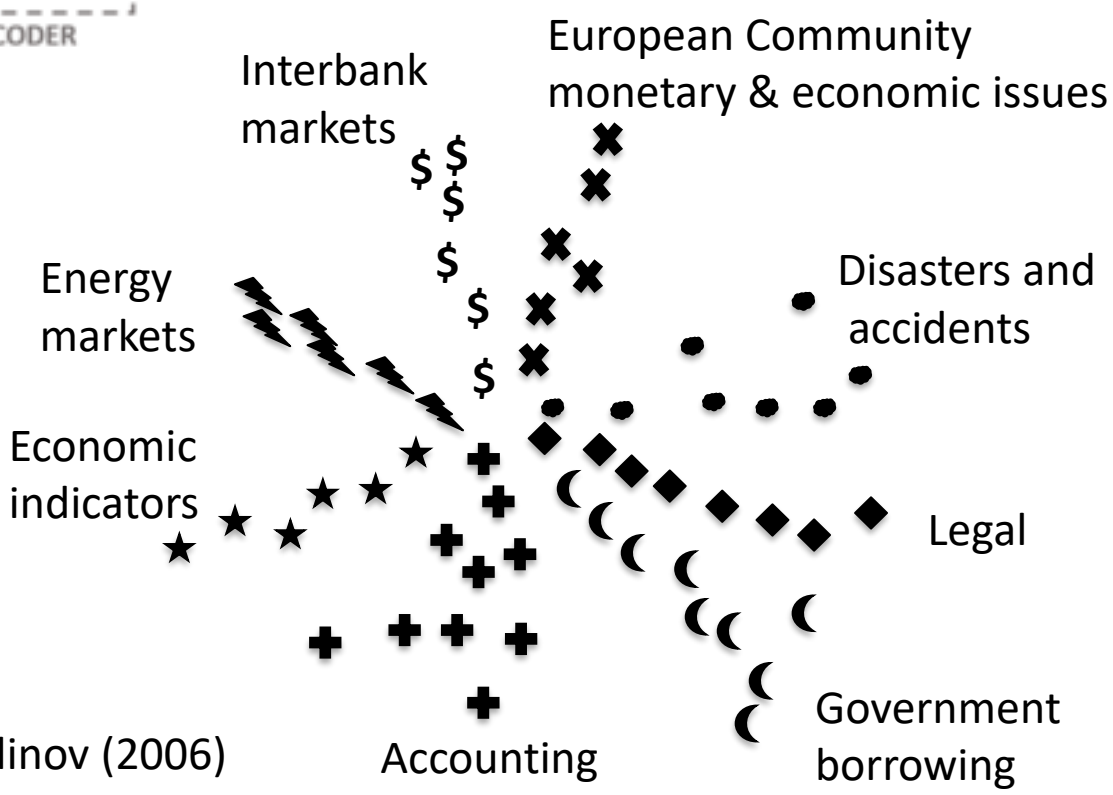


Hidden layer has less neurons than input/output layers (often called „bottleneck")

Autoencoder

PCA

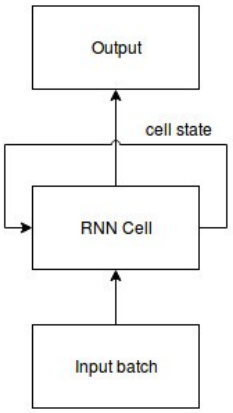




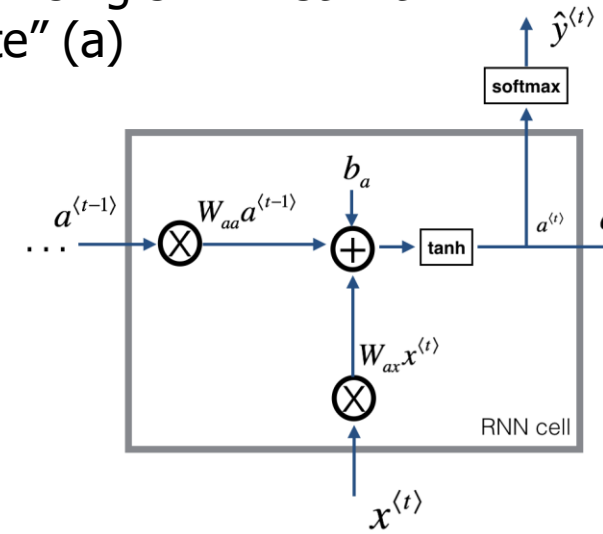
Adapted from Hinton and Salakhutdinov (2006)

RNN - Recurrent neural networks

- **Feed forward** nets have no memory. Each pass is independent.
- **RNN** are networks with loops, allowing information to persist.
- **RNN** connects previous information to the present task. This previous information is embedded in „state" (memory) of the basic unit – cell.
- **RNN** is suitable for prediction of sequential data (text, videos etc.).



Cell = single NN net with „state" (a)



a = state vector
b = bias

$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

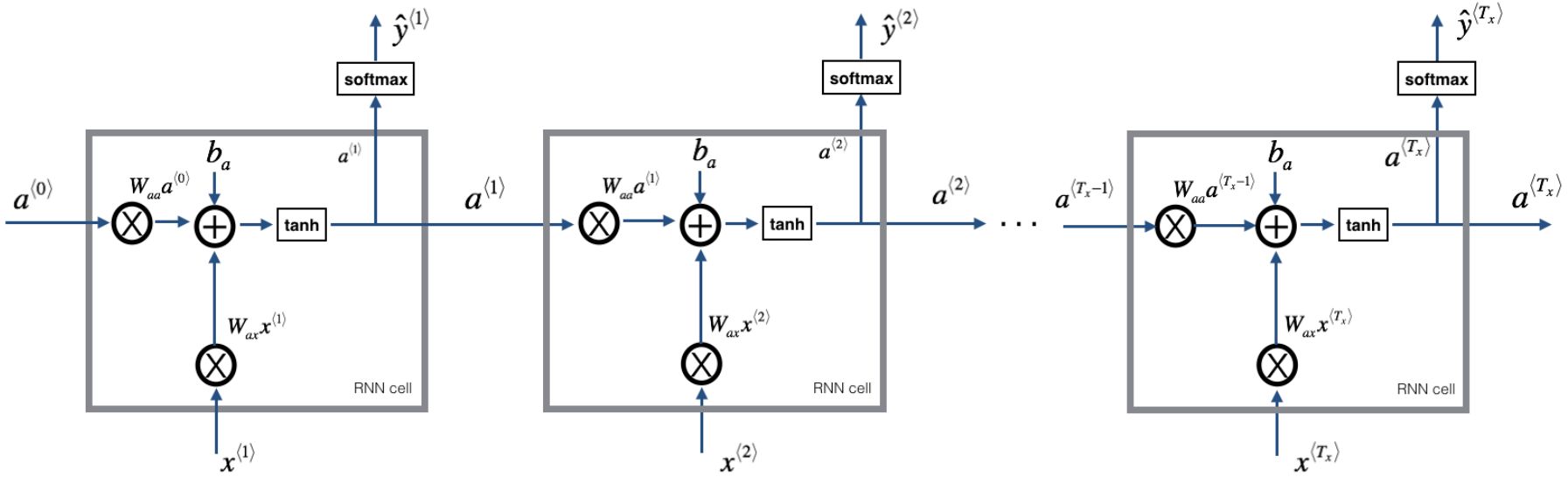
$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

Ref.: https://github.com/Kulbear/deep-learning-coursera

RNN - Recurrent neural networks

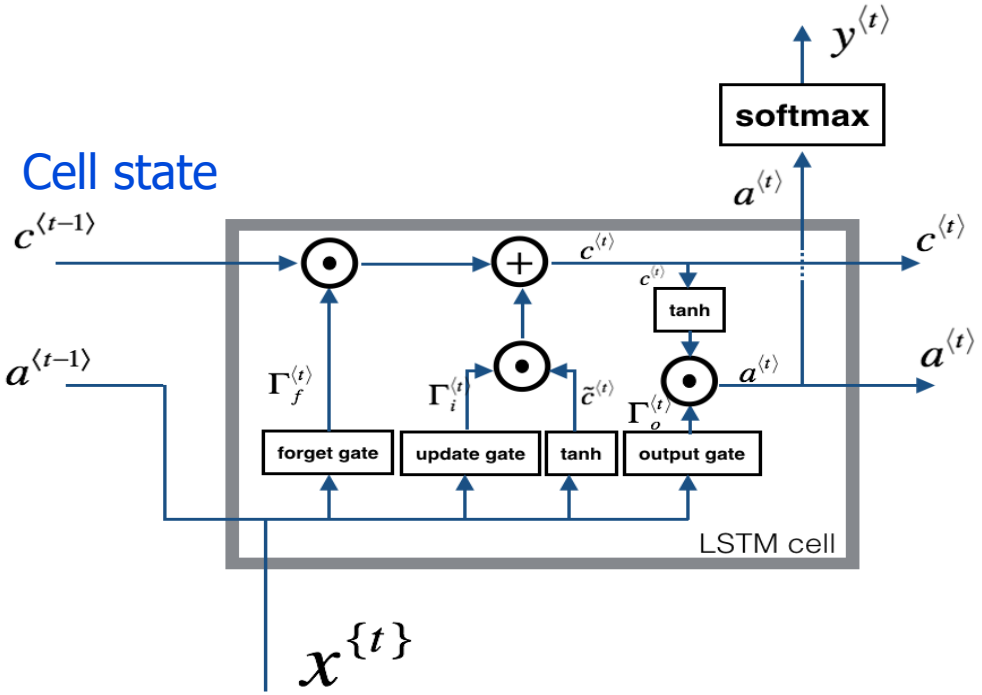
- During processing of input sequences **RNN** can be thought of as multiple copies of the same network, getting as input different vector at each timestep (a.k.a. unrolled RNN).
- Number of timesteps is given when building **RNN**.
- Each network copy passes a message (state „a") to a successor.
- Parameters (weghts) values are shared (the same) in each network copy.
- If the gap of relevant information and the point where it is needed becomes large **RNN** is impractical to realize (short memory only).

Unrolled RNN



LSTM – a special kind of RNN

- **LSTM** (Long Short Term Memory) can learn long term dependances.
- Instead of a single NN layer in a cell there are four layers with sigmoid and tanh activations called „gates" (forget, update, tanh, output).
- **LSTM** has a more complex cell state „c" which act as a conveyor belt between unrolled cells.
- **LSTM** can add or remove information from the cell state „c" by processing previous simple state „a" through gates.



Single cell processing

$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

$$\tilde{c}^{\{t\}} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$

$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$

$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Deep Learning

Tools for deep learning (Hidden prerequisite: Convolution and Backpropagation in vector matrix form → **GPU**)

Theano

- A free library in Python which has been developed with the specific goal of facilitating research in deep learning
- It is also a powerful general purpose tool for general mathematical programming
- Theano extends NumPy (the main Python package for scientific computing) by adding symbolic differentiation and GPU support, among various other functions
- It provides a high-level language for creating the mathematical expressions that underlie deep learning models, and a compiler that takes advantage of deep learning techniques, including calls to GPU libraries, to produce code that executes quickly
- Theano supports execution on multiple GPUs

Deep Learning

Tensor Flow

- C++ and Python based software library for the types of numerical computation typically associated with deep learning.
- It is heavily inspired by Theano, and, like it, uses dataflow graphs to represent the ways in which multidimensional data arrays communicate between one another.
- These multidimensional arrays are referred to as "tensors."
- Tensor Flow also supports symbolic differentiation and execution on multiple GPUs.
- Designed by Google.
- It was released in 2015 and is available under the Apache 2.0 license.

Deep Learning

Torch

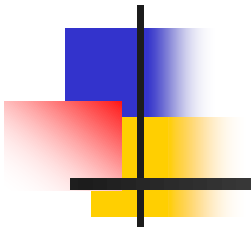
- An open-source machine learning library built using C and a high-level scripting language known as Lua.
- It uses multidimensional array data structures and supports various basic numerical linear algebra manipulations.
- It has a neural network package with modules that permit the typical forward and backward methods needed for training neural networks.
- It also supports automatic differentiation.
- PyTorch based on Torch (June 2020.).

Deep Learning

Lasagne, Keras and cuDNN

- Lasagne is a lightweight Python library built on top of Theano that simplifies the creation of neural network layers
- Similarly, Keras is a Python library that runs on top of either Theano or TensorFlow that allows one to quickly define a network architecture in terms of layers and also includes functionality for image and text preprocessing
- cuDNN is a highly optimized GPU library for NVIDIA units that allows deep learning networks to be trained more quickly. It can dramatically accelerate the performance of a deep network and is often called by the other packages above.

- **Deeplearning4j** - Java-based open-source deep learning library
- **Caffe** - C++ and Python based BSD-licensed convolutional neural network library
- **Cognitive Toolkit (CNTK)** – Microsoft: Python, C#, C++



Osvrt na modele ANN za duboko učenje

(Izvor: IEEE Spectrum, October 2021.)

Osvrt na modele za duboko učenje

- Umjetne neuronske mreže (ANN) su fleksibilni računalni model – univerzalni aproksimatori funkcija. Mogu se prilagoditi raznim tipovima podataka. Počeci sežu s kraja 1950-tih.
- Tipični model: višeslojni perceptron (engl. Multi Layer Perceptron – MLP), ali s malim brojem slojeva (tipično 3).
- ANN bile su vrlo popularne i raširene do početka 2000. godine , kada su pale u zaborav. Razlog: nedovoljni računalni resursi za izračunavanje velikog skupa parametara (t.j. težinskih faktora svakog ulaza u neuron) za veći broj slojeva.
- **Duboke mreže** su moderna inkarnacija ANN (početkom 2012. god. uporabom matričnog izračunavanja i GPU-ova) s mnogo više ulaza, izlaza i međuslojeva potencijlno spojenih na tisuće načina.
- Posljedično sadrže veliki broj parametara (težina) koji se moraju podesiti učenjem.
- **Fleksibilnost i složenost mreža za duboko učenje povlači ogroman trošak računalnih resursa.**

Osvrt na modele za duboko učenje

- Iz teorije statističkih modela slijedi da poboljšanje performansi za faktor k traži najmanje k^2 podataka za učenje, a zbog velikog broja internih parametara to se penje na najmanje k^4 .
- Pregledom oko 1000 objavljenih radova uočeno je stvarno skaliranje izračunavanja **na 9-tu potenciju**. To bi značilo da smanjenje pogreške na pola traži 500 puta veće računalne resurse.
- Napredak u izvedbi sklopovlja (Moore-ov zakon i uporaba GPU jezgri) donosi samo manja poboljšanja. I dalje se traži velik broj procesora i dulje vrijeme učenja mreže.
- Primjer: Google DeepMind
  - *AlphaGo - Lee* (Konvolucijska mreža); 1920 procesora, 280 GPU-a, 64 niti izvođenja, više mjeseci učenja. Cijena 35 M$. Kasnije poboljšano uvođenjem specijaliziranih sklopova (TPU – Tensor Processing Units)
  - *StarCrft II* video igrica, odustalo se ispitivati različite arhitekture mreže zbog prevelike cijene učenja.

Osvrt na modele za duboko učenje

- Klasifikacija slika na standardnom skupu mrežom AlexNet (2012. god.), dva GPU, učenje 6 dana. Isti problem 2018. god mreža NASNet-a smanjila je pogrešku AlexNet-a na pola ali uz 1000 puta više izračunavanja.
- Ekstrapolacijom na sličnim primjerima slijedi da cijena poboljšanja performansi mreža za duboko učenje uskoro (2 do 3 god. ?) postaje neprihvatljiva i tehnički neodrživa.
- To je tipičan životni ciklus svake tehnologije („nema srebrnog metka”).
- Moguća rješenja:
  - Specijalizirani sklopovi (slično kao TPU-ovi), ali uporabom se gubi generalizacija.
  - Novi, drugačiji, sklopovi opće namjene (analogni, optički, kvantni, …).
  - Optimiranje i balansiranje cijene generiranja (implementiranja) mreže i troškova učenja uporabom manjih mreža.
  - Jedna mreža za više zadataka (a.k.a. meta-learning) ali uz povećanje pogreške.
  - Sinergija mreža za duboko učenje i ekspertnih sustava.
  - . . .