Gépi tanulás

Nyelv és informatika – Pécs, 2022/23 tavasz
6. óra

Simon Eszter – Vadász Noémi

2023. április 1.

Tartalom

Gépi tanulás

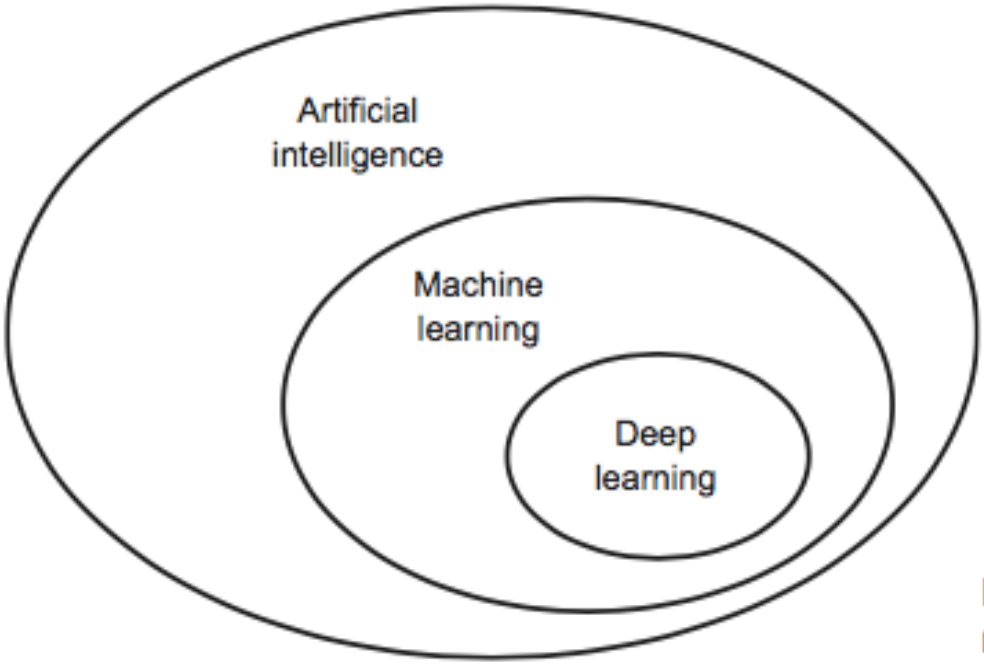


Figure 1.1 Artificial intelligence, machine learning, and deep learning

A hagyományos gépi tanulás

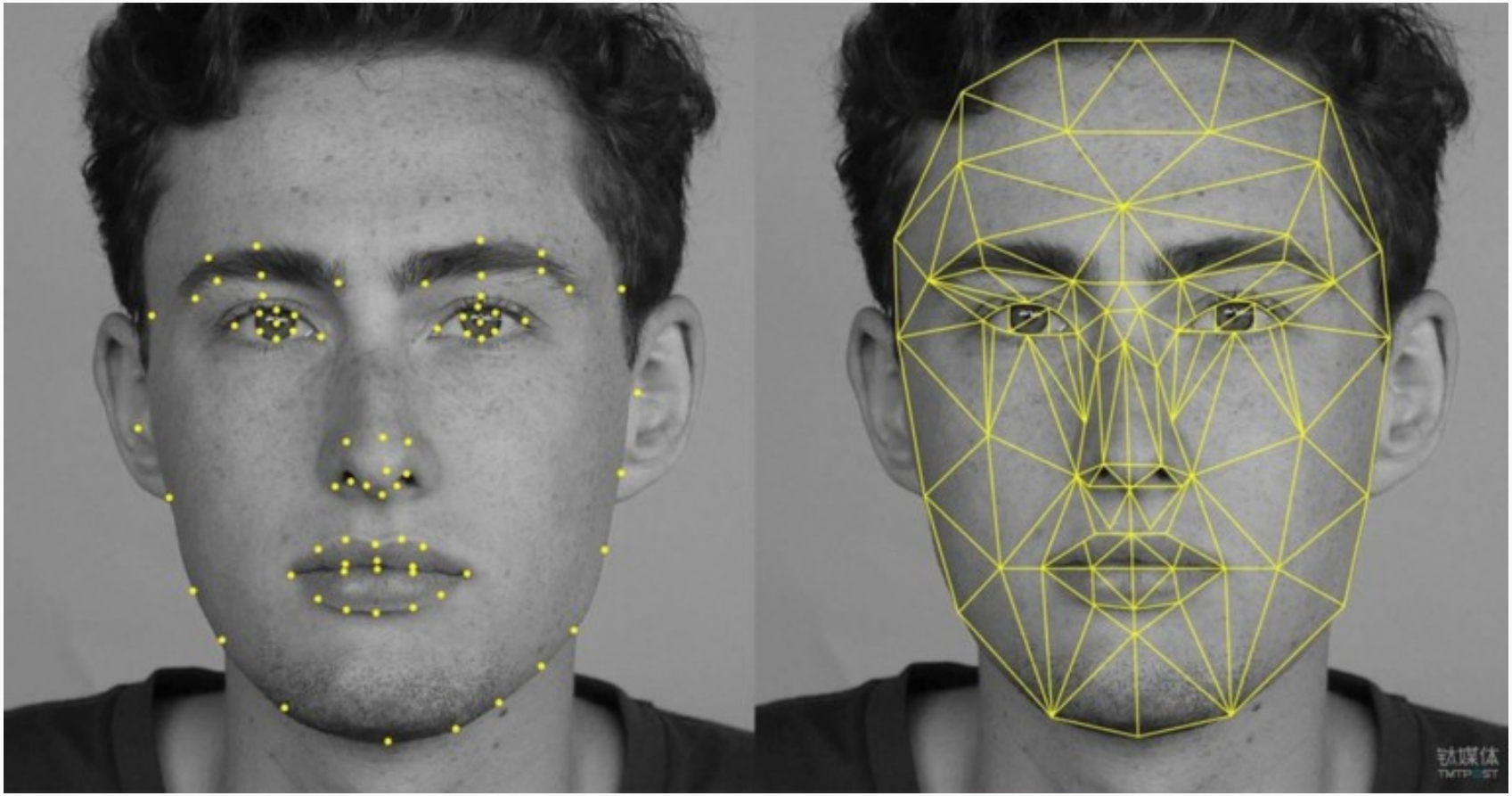
Módszerek (ismétlés)

- szabályalapú
- statisztikai, sztochasztikus, hagyományos gépi tanulás, machine learning
- neurális, mélytanulás, deep learning

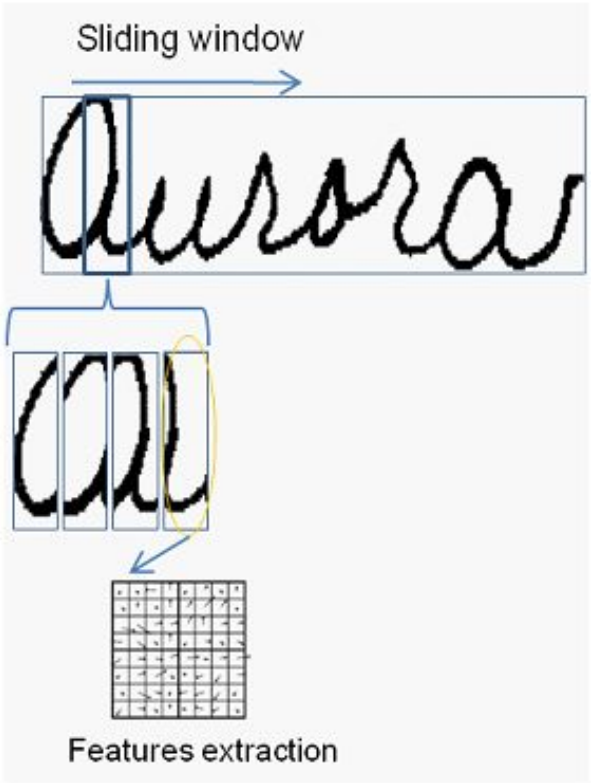
Gépi tanulás

- már az ötvenes években
- az ezredforduló környékén, de főleg a 2010-es évesben terjedt el
- pl. képek elemzése, arcfelismerés, gazdasági előrejelzések, önvezető autók, ajánlórendszerek, az ipar sok területén
- gépi tanulás != magolás (a magolás könnyű feladat lenne a számítógépnek, a valódi tanulás: az új szituációkra való *általánosítás* feladata) véges sok tanítási minta segítségével meg kell adni egy függvényt, ami kapcsolatot teremt az adatok között, de végtelen sok lehetőségből kell választani

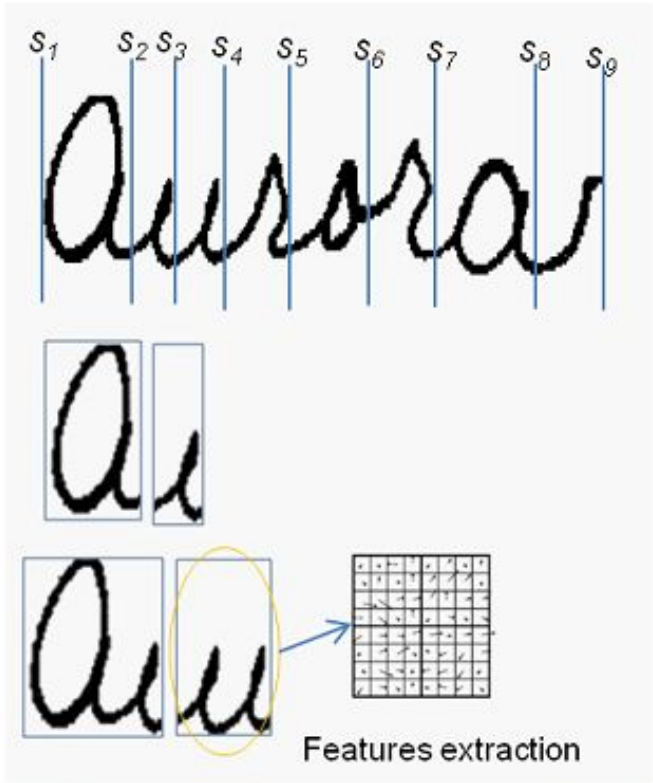
Arcfelismerés



Karakterfelismerés



(a)Segmentation-free method



(b)Over-segmentation-based method

A statisztikai alapú rendszerek

előnyei:

- az egyes elemzésekhez valószínűségek kapcsolódnak, az elemzések a valószínűségük alapján rangsorolhatók
- akkor is adhat jó eredményt, ha a mögöttes nyelvmodell nem adekvát
- flexibilis megközelítés a szabályalapúhoz képest

hátrányai:

- nagy mennyiségű annotált adatot igényel
- a rendszer alkalmazása más nyelvre, doménre nagy teljesítménybeli visszaesést okozhat

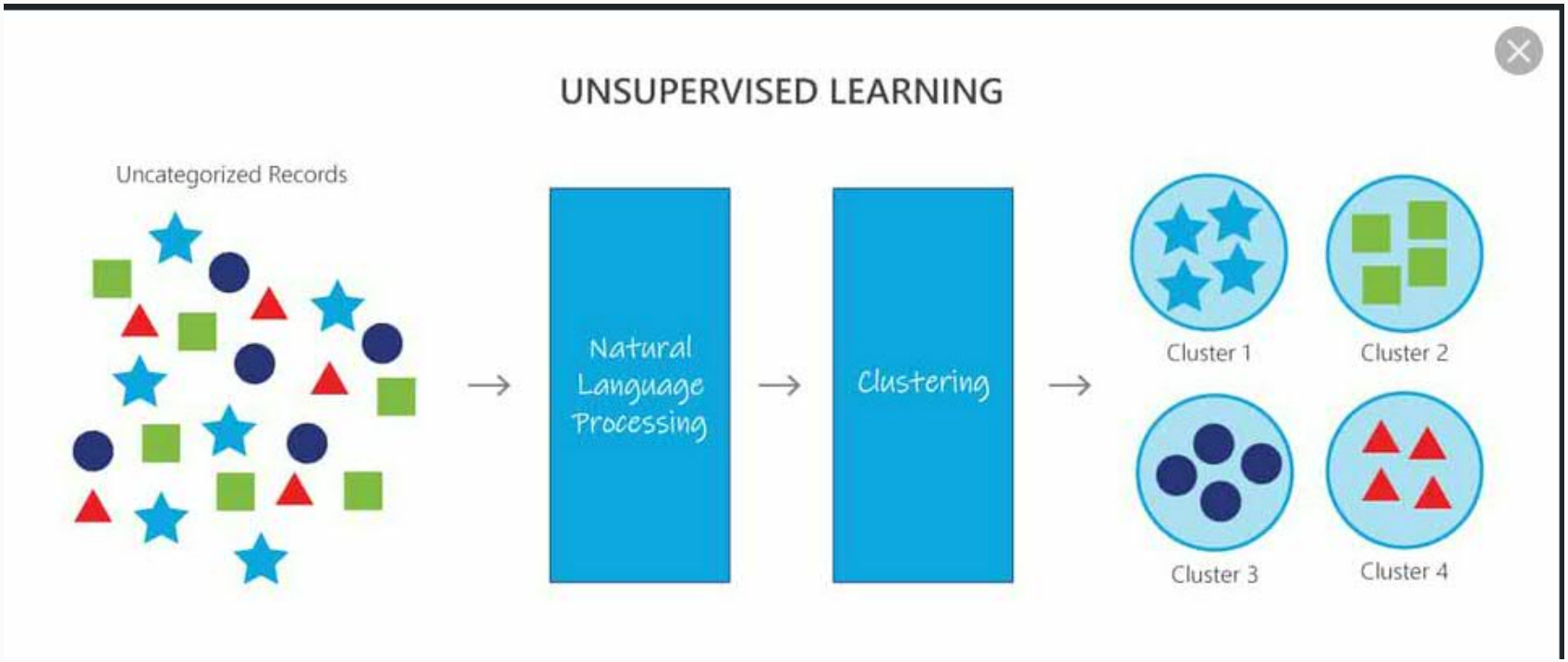
Felügyelet nélküli gépi tanulás

bemenet: nyers szöveg

De mit lehet megtanulni a nyers szövegből? A felügyelet nélküli tanulás olyan gépi tanulási feladat, ahol a cél jelöletlen/címkézetlen adat leírása rejtett struktúrák/összefüggések feltárásával.

- elsősorban az adatmegértés a cél
- klaszterezés: egy adatbázis címkézetlen egyedeinek olyan csoportjainak megtalálása felügyelet nélküli tanulási keretben, hogy az egy csoportban levő egyedek hasonlóbbak lesznek egymáshoz, mint a más csoportban levőkhöz
    1. nincsenek előre definiált osztályok, ha új típusokat akarunk találni
    2. előre megszabjuk az osztályok számát, ha egy bizonyos feladatban megszokott osztályokat keressük

A klaszterezés



Felügyelt gépi tanulás

- feltételezés 1.: az adatpontok egymástól független elemek, amelyeknek egyenletes az eloszlásuk
- feltételezés 2.: az eddig nem látott adatpontokra is igaz a fenti állítás
- $\rightarrow$ a már látott nyelvi elemekből tudunk következtetni a még nem látottakra
- a nyelvtechnológiában: az annotált korpuszból tanulja ki a számítógép az adatpontokra jellemző jegyeket (majd annotált korpuszon is értékelünk ki)

A felügyelt gépi tanulás forgatókönyve

A felügyelt gépi tanulás forgatókönyve (címkézési feladat)

1. gold standard korpusz
2. train-devel-test halmaz
3. jegykinyerés
4. modellépítés
5. prediktálás (taggelés)
6. kiértékelés

train-devel-test

- a gold standard korpuszt felosztjuk halmazokra
  1. train: ezen tanítunk
  2. development: ezen fejlesztünk
  3. test: ezen értékelünk ki
- a teszthalmaz elemei nem szerepelhetnek a tanítóhalmazban!
- ez egyrészt csalás, másrészt a rendszer túlzottan ‘rátanul’ a szövegre, nem lesz képes az általánosításra

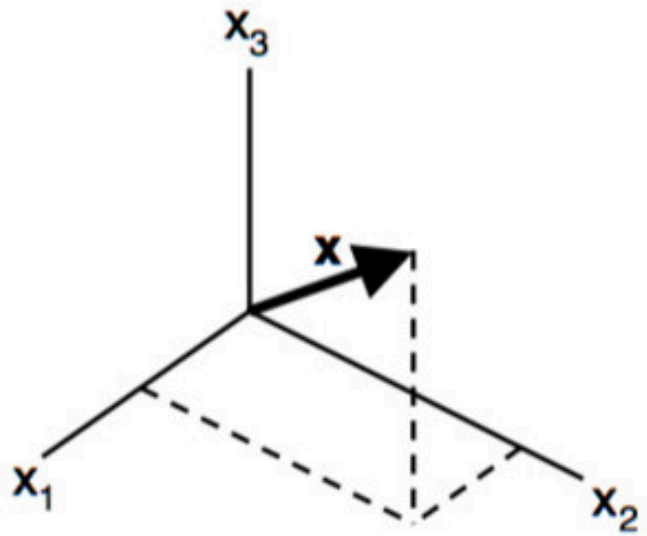
Jegykinyerés (feature extraction)

- a jegyek az adatpontok különféle tulajdonságait írják le
- a jegyeket a számítógépes nyelvész találja ki, definiálja és kódolja
- a jegy hasznosságát az adat határozza meg: a jegy megkülönböztető erejét ki kell mérni, utána eldönteni, hogy alkalmazzuk-e
- a jegyek hozzáadása vagy a paraméterek állítása egyesével, majd mérés, ha nem ront, akkor meghagyjuk
- a jegyek vektorokra képeződnek le

Jegykinyerés (feature extraction)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector



Feature space (3D)

Jegytípusok (HunTag)

forrás: felszíni tulajdonságok, morfológiai információk, szintaktikai információk, listatagság

érték: sztring, bináris

egység: token, mondat

pl. bináris felszíni jegyek: *hascap, allcaps, capperiod, camel, 3caps, iscap*

pl. morfológiai jegyek: *lemma, fulltag, pos, tagend, oov, tagpattern, isbetweensamecases, plural*

pl. szintaktikai jegyek: *sentstart, sentend, NpPart, parsePatts*

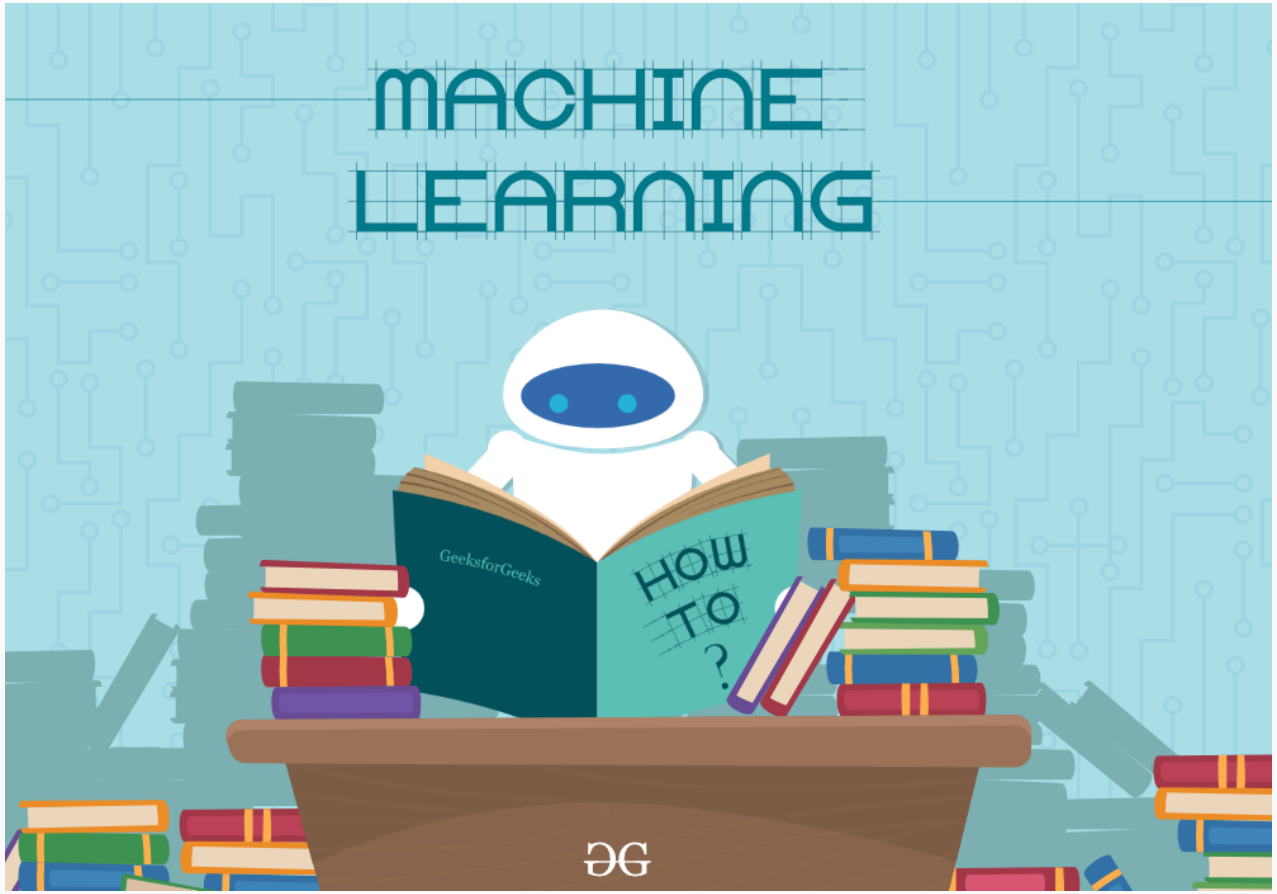
pl. listatagság: is-a reláció, pl. Budapest benne van a városok listájában, akkor város

Jegyek a névelemfelismerésben

- Ortográfiai jellemzők
  kezdőbetű típusa, szóhossz, tartalmaz számot/írásjelet, arab/római szám
- Gyakorisági adatok
  kis/nagybetűs-, mondatközi nagybetűs/nagybetűs arányok, gyakoriság
- Szövegkörnyezet info trigger uni-/bi-/trigramok, mondatpozíció, dokumentumon belüli pozíció
- Kifejezésszintű információ
  megelőző tokenek címkéi, zárójelben/idézőjelben van, reguláris kifejezések
- Egyértelmű szavak szótára
  tanuló adatbázisból összegyűjtve, pl. betegségek nevei
- Trigger szótárak
  keresztnevek, országok, városok...

Modellépítés

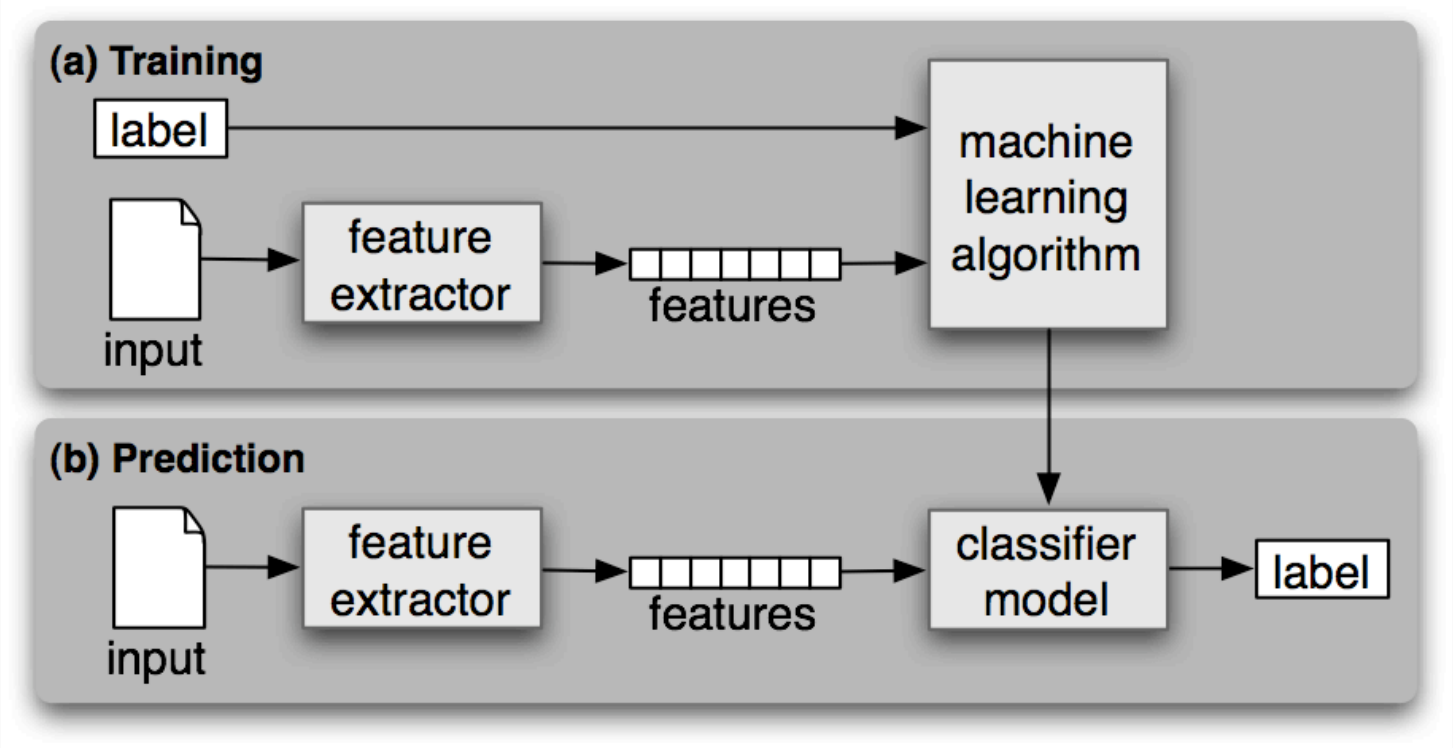
- a jegy-címke párokhoz súly van hozzárendelve, ami azt mutatja meg, hogy az adott jegy mennyire van hatással arra, hogy az adott jeggyel rendelkező token az adott címkét kapja



Prediktálás és kiértékelés

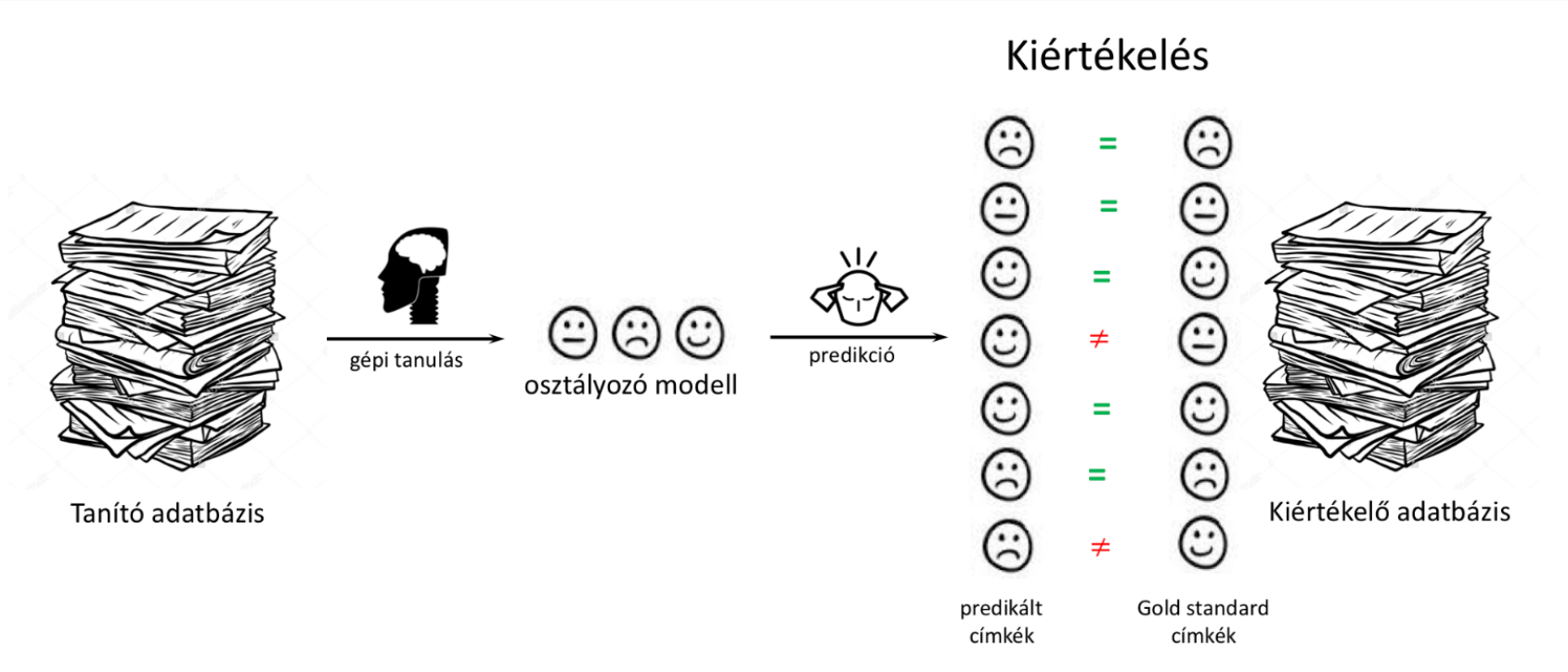
- a teszthalmazon!
- a teszthalmaz feature-izálása, majd a feature-vektorok alapján a címkék kibocsátása
- az egyes tokenekhez azok a címkék kerülnek kiosztásra, amik a jegyvektorok alapján a legnagyobb valószínűséget kapták
- az eredményt összevetjük a teszthalmaz gold-standard címkéivel
- pontosságot, fedést, F-mértéket számolunk
- a rendszerünk készen áll arra, hogy további szövegeket címkézzünk vele (várható pontossággal, fedéssel) :)

Áttekintés


(a) Training
label
input
feature extractor
features
machine learning algorithm
(b) Prediction
input
feature extractor
features
classifier model
label

Kiértékelés

Honnan tudhatjuk, hogy egy erőforrás vagy eszköz jó?



Accuracy: az összes predikció közül hány esetben értett egyet a program a kiértékelő adatbázisban szereplő címkével?

Képeket mutatunk, a program jelzi, ha macskát lát.

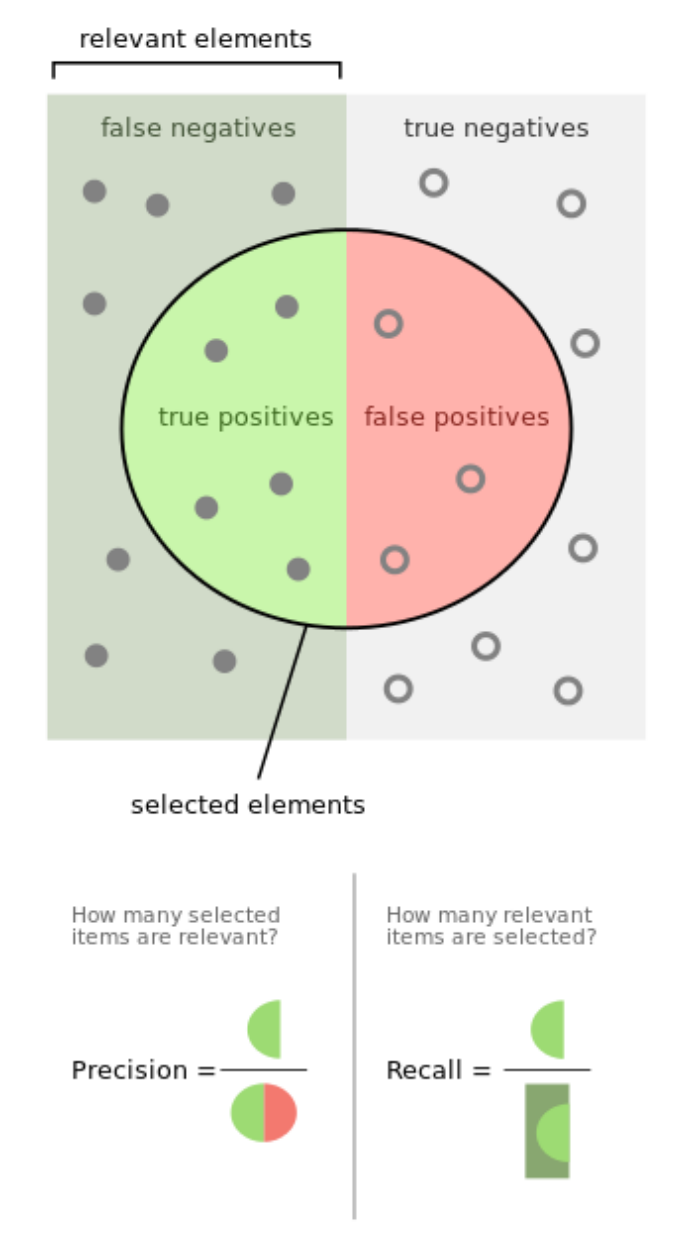
- TP: hit
  macskát mutattunk, macskát mondott
- FN: type II errror, miss, underestimation
  macskát mutattunk, nem mondott semmit
- FP: type I error, false alarm, overestimation
  kutyát mutattunk, macskát mondott
- TN: correct rejection
  kutyát mutattunk, nem mondott semmit

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Pontosság és fedés



A “jóság” mérőszámai

Hogy torzíthatjuk az eredményeket?

- a **pontosság** (precision, ami azt mutatja, hogy a találatokból hány volt eredetileg jó) növelésére: a program sose mond semmit
- a **fedés** (recall, ami azt mutatja, hogy az eredetileg jók közül hányat találtunk meg) növelésére: a program mindig macskát mond

Ellenszer: **F-mérték** (F-measure, F-score): a pontosság és a fedés harmonikus közepe, átlaga

Pontosság, fedés, F-mérték

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}$$

F-mérték

- a *pontosság* maximalizálása: minél kevesebb tévedés $\rightarrow$ szigorítás
- a *fedés* maximalizálása: minél több találat $\rightarrow$ megengedőbb rendszer

$\beta = 1$

$$F = 2 \times \frac{precision \times recall}{precision + recall}$$