Üzleti elemzések módszertana

Gyakorlat

Kuknyó Dániel

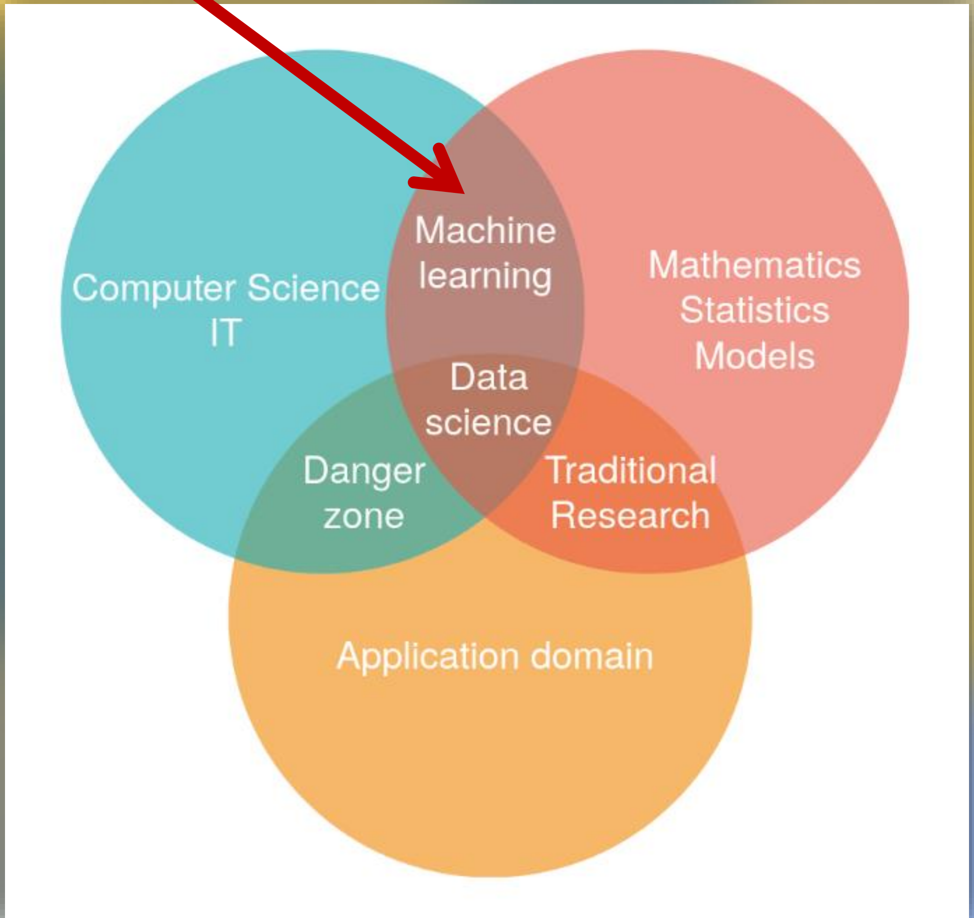
daniel.kuknyo@mailbox.org

18 57

BGE

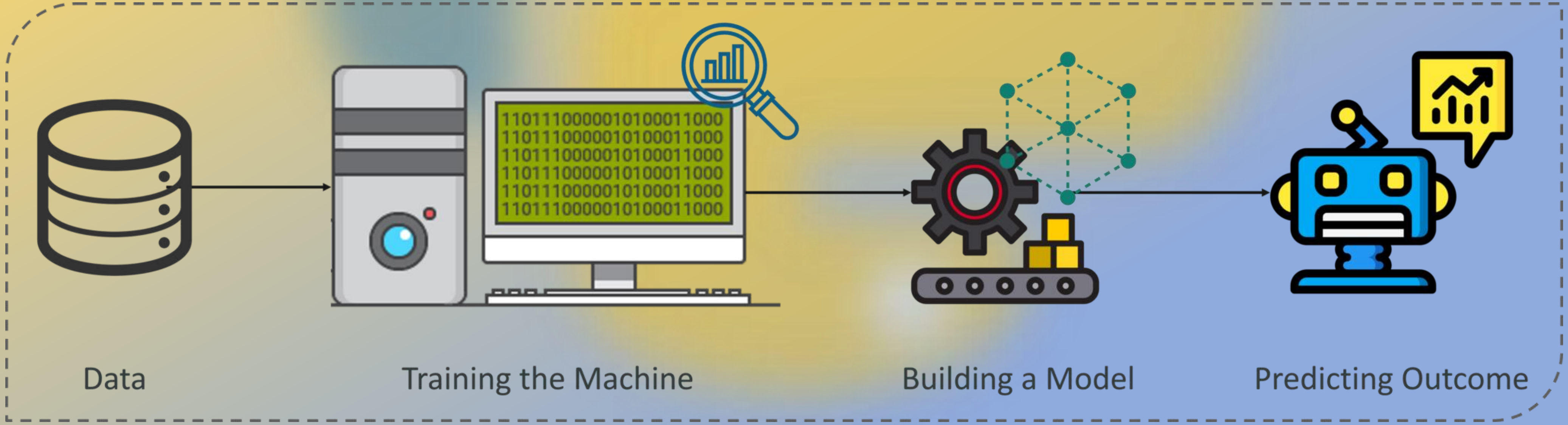
1. Előadás
Alapfogalmak
Lineáris regresszió
Gradiens ereszkedés

Hol vagyunk?


Machine
learning
Computer Science
IT
Mathematics
Statistics
Models
Data
science
Danger
zone
Traditional
Research
Application domain

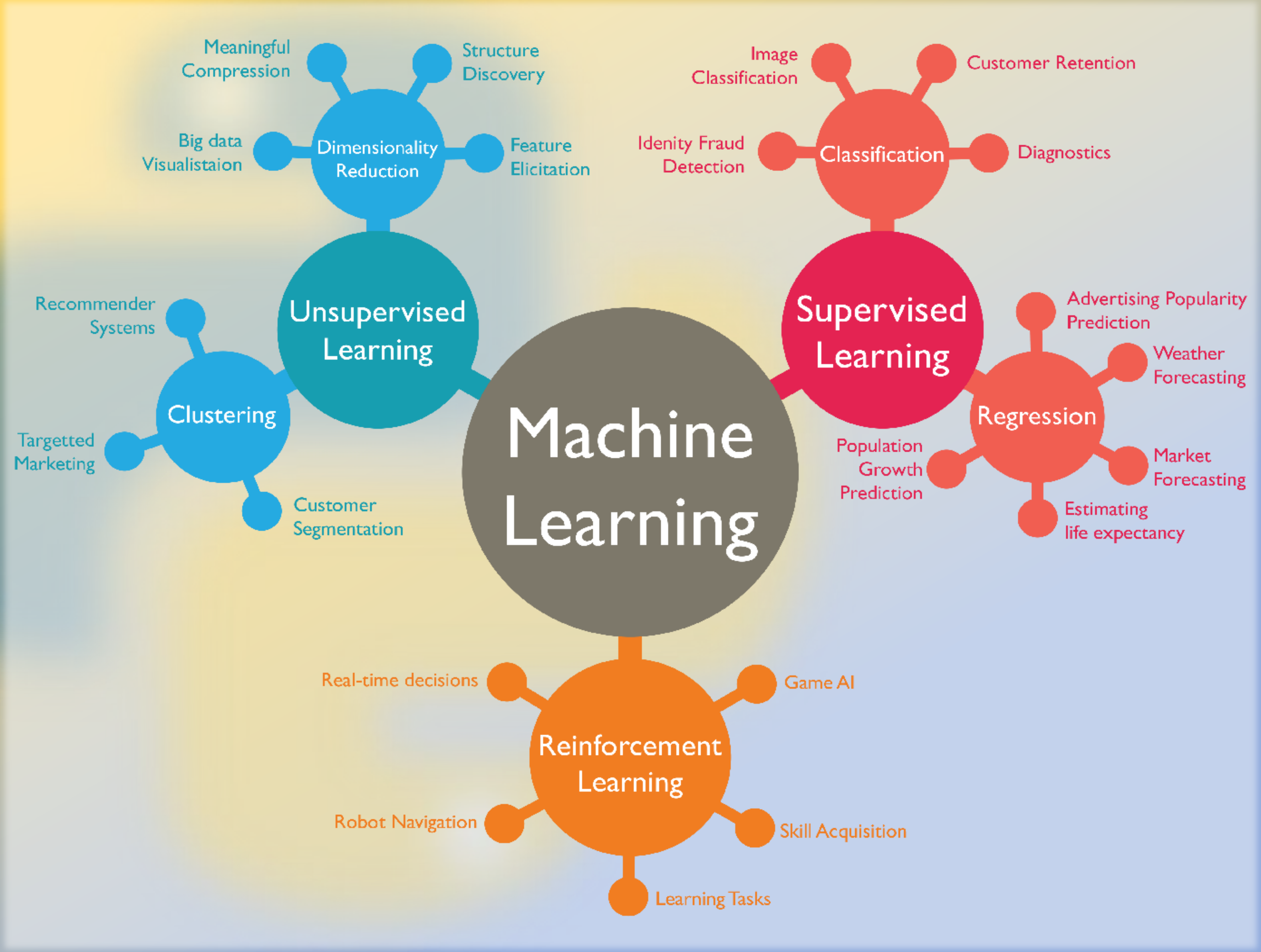
A megfontolás az ML mögött

- A hagyományos szemléletmódban utasításokat írtunk egymás után, ciklusokban, függvényekben…
- A gépi tanulás szemléletmódjában explicit programozás nélkül tanítjuk meg a számítógépnek (robotnak?), hogy mit tegyen.
- Ez hogyan történik? A gép tapasztalat alapján tanul!



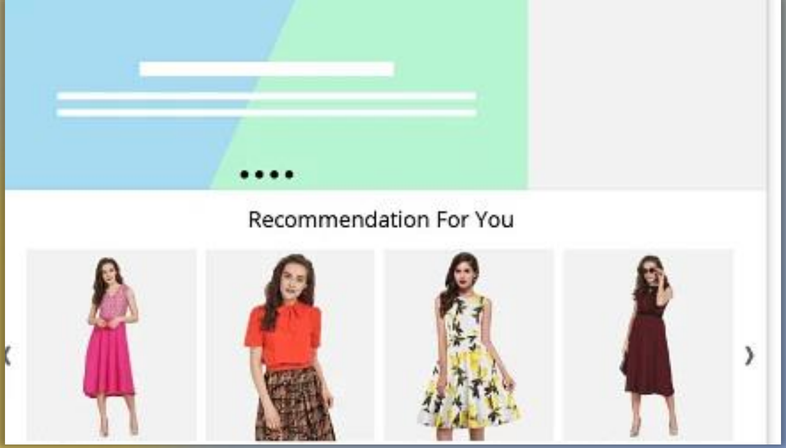
Az ML térkép

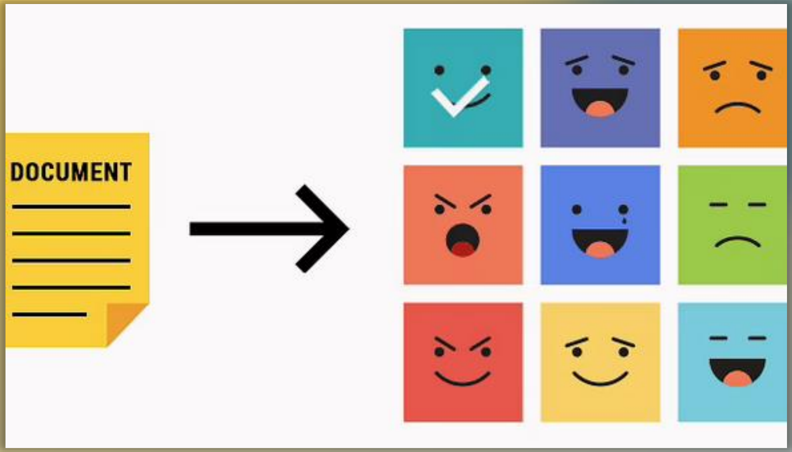
- Felügyelt tanulás
  - Regresszió
  - Osztályozás
- Felügyelet nélküli tanulás
  - Dimenziócsökkentés
  - Klaszterezés
- Megerősített tanulás
  - Robot utasítás
  - Valós idejű döntések
  - Ágens tanítás

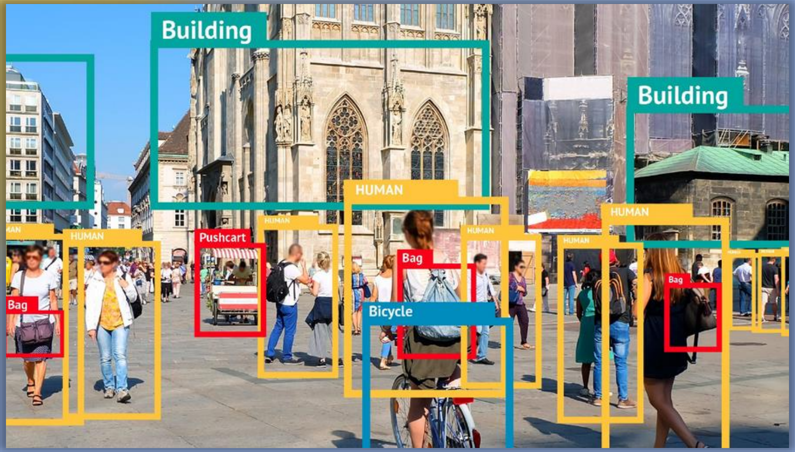


Mire jó a gépi tanulás?

- Objektumok felismerése
- Jelentések elemzése
- Javaslatok készítése
- Hangfeldolgozás
- Képfeldolgozás
- Robotok vezérlése
- Orvosi alkalmazások
- Biztosítási kreditek



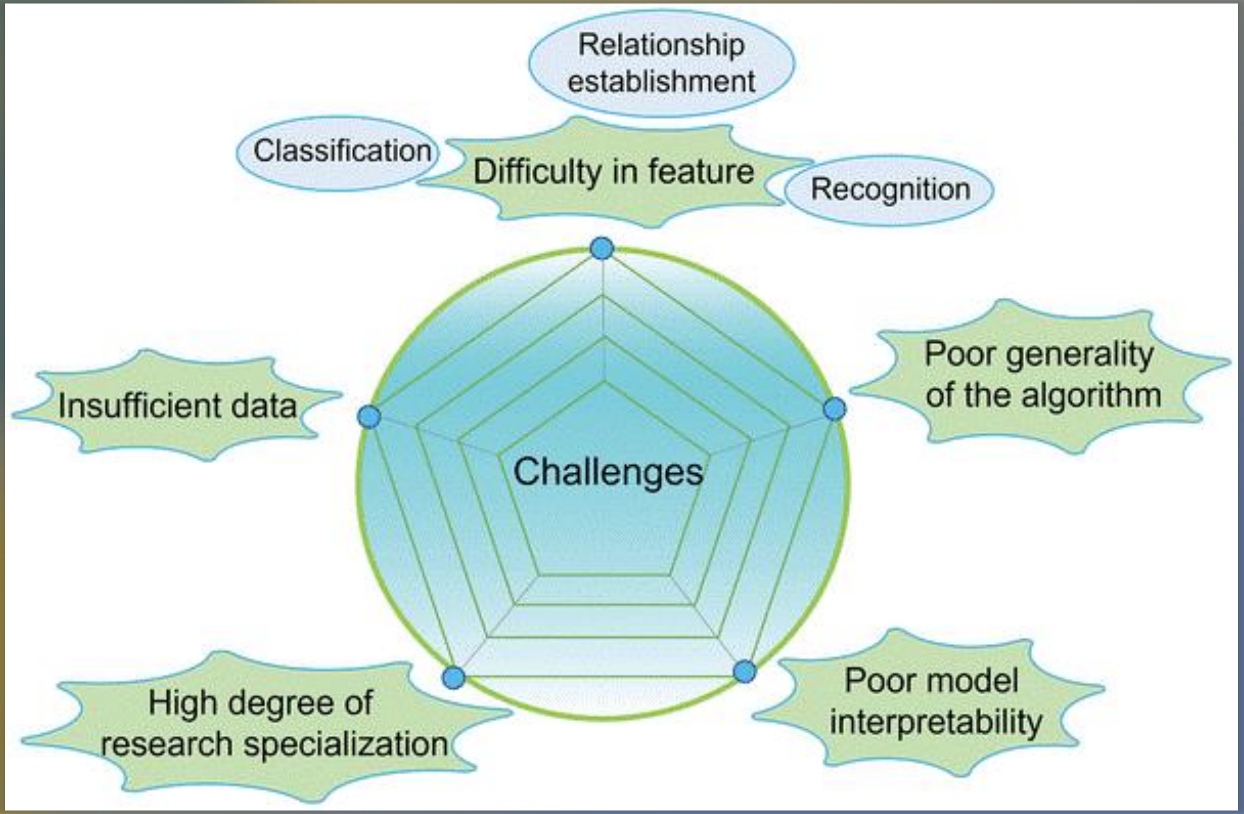




- Röviden: a körülöttünk lévő világ értelmezésére.

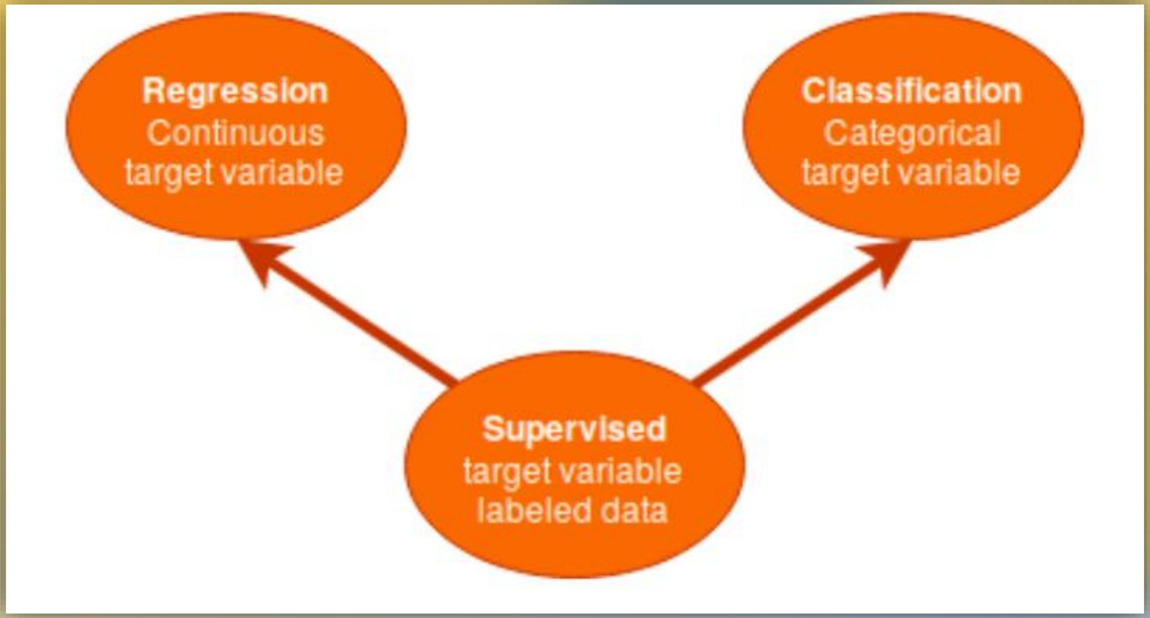
Kihívások a gépi tanulás területén

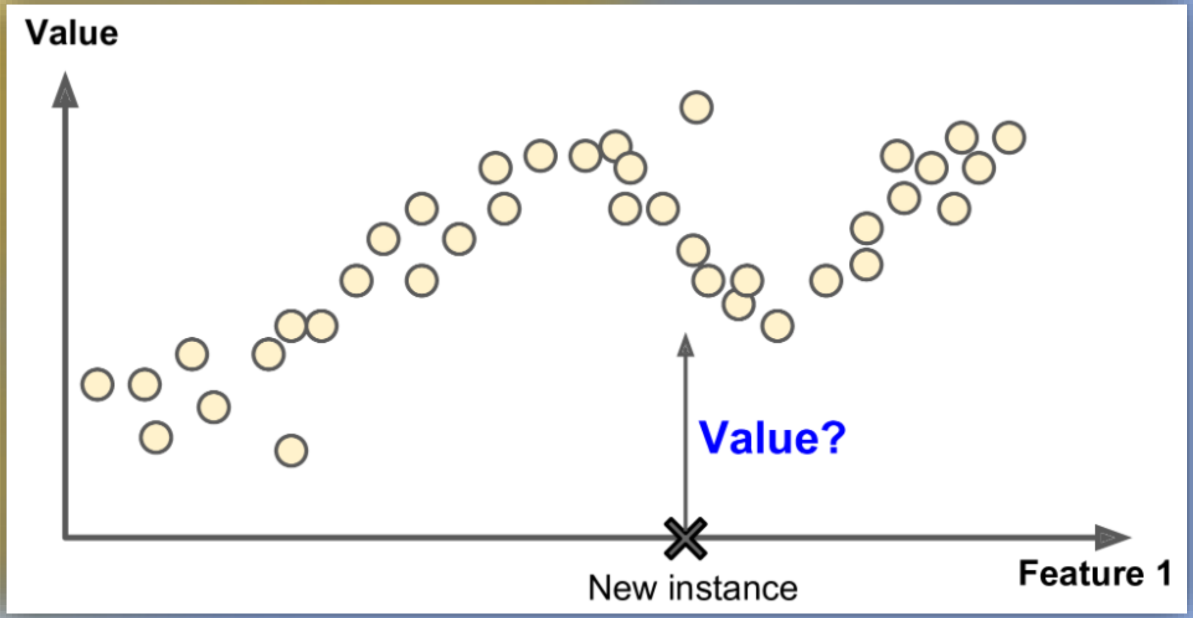
- Gyenge minőségű adatok.
- Nem reprezentatív adatok (sampling bias).
- Felesleges jellemzők.
- Gép és ember kapcsolata.
- Folyamatosan változó világ.
- A problémák eredhetnek:
  - Rossz változókból
  - Rosszul általánosító algoritmusból
  - Nehezen értelmezhető eredményekből
  - Nagyon specifikus szakterületi specializációból
  - Elégtelen minőségű / mennyiségű adatból



A regresszió

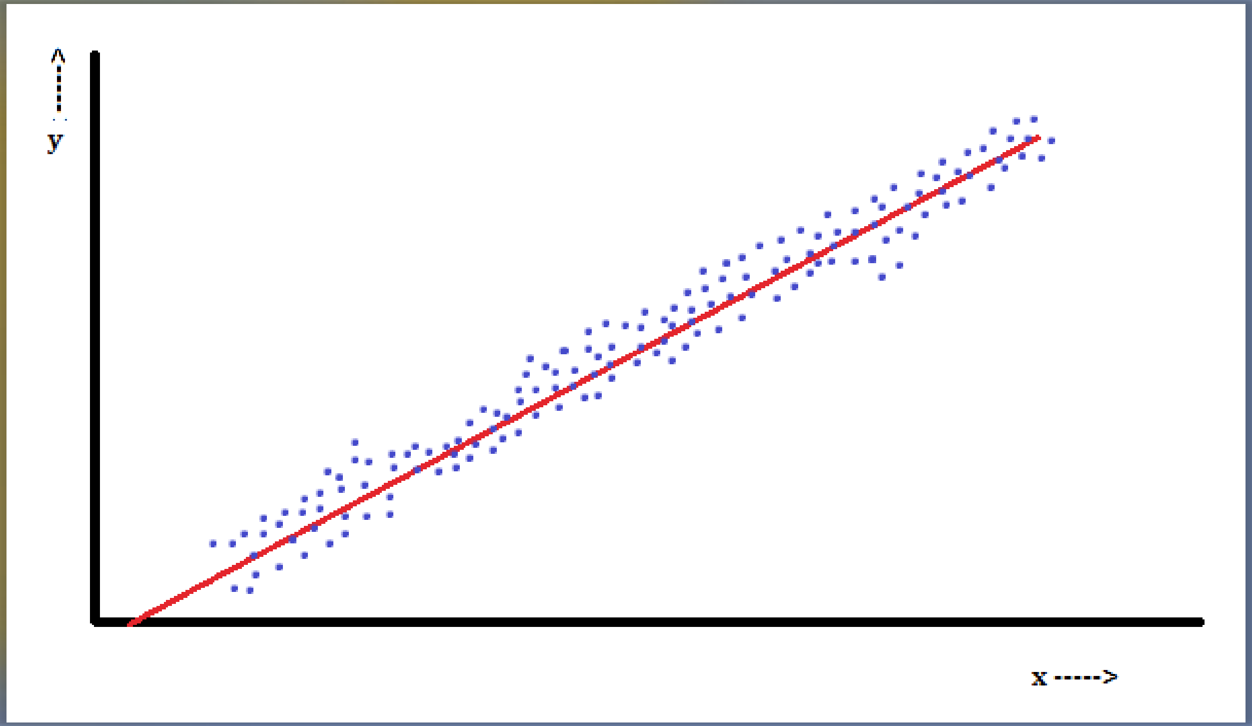
- Statisztikai elemző eljárás, amely változók közötti kapcsolatot modellez.
- Eredménye a **modell**.
- **Folytonos** változóra vonatkozik.
- Felügyelt tanítás kategóriájába tartozik: ismertek a címkék.





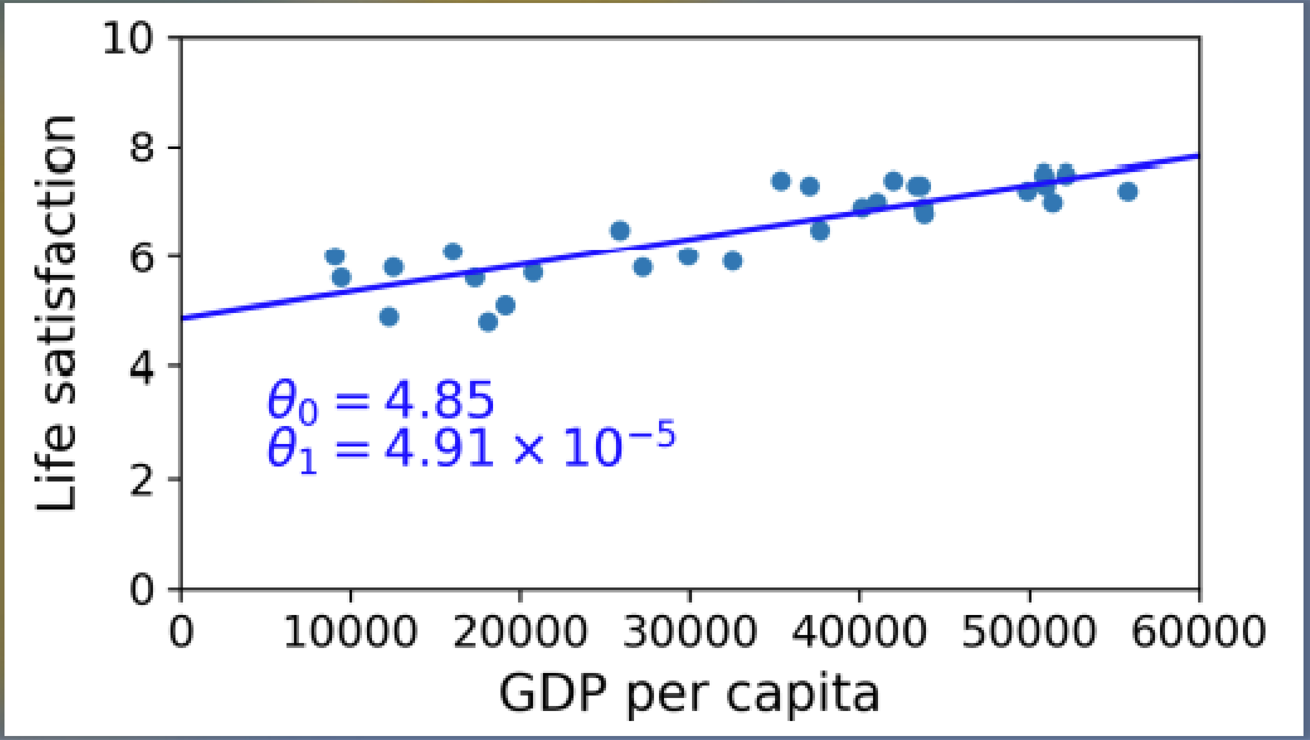
A regresszió elemei

- A regressziós elemzésben adott egy jelenség, ami az előrejelzés tárgyát képezi. Lehet pl. házak ára, emberek fizetése stb.
  Ennek a neve: **célváltozó**, függő változó, válasz, output.
- Adottak még megfigyelések, amelyek alapján a becslést végezzük.
  Minden megfigyeléshez tartozik legalább két jellemző.
- Ezeket a jellemzőket nevezzük független változónak, inputnak vagy **prediktornak**.
- A képen mi a megfigyelés?
  Mi a modell?
  Mi a jellemző?



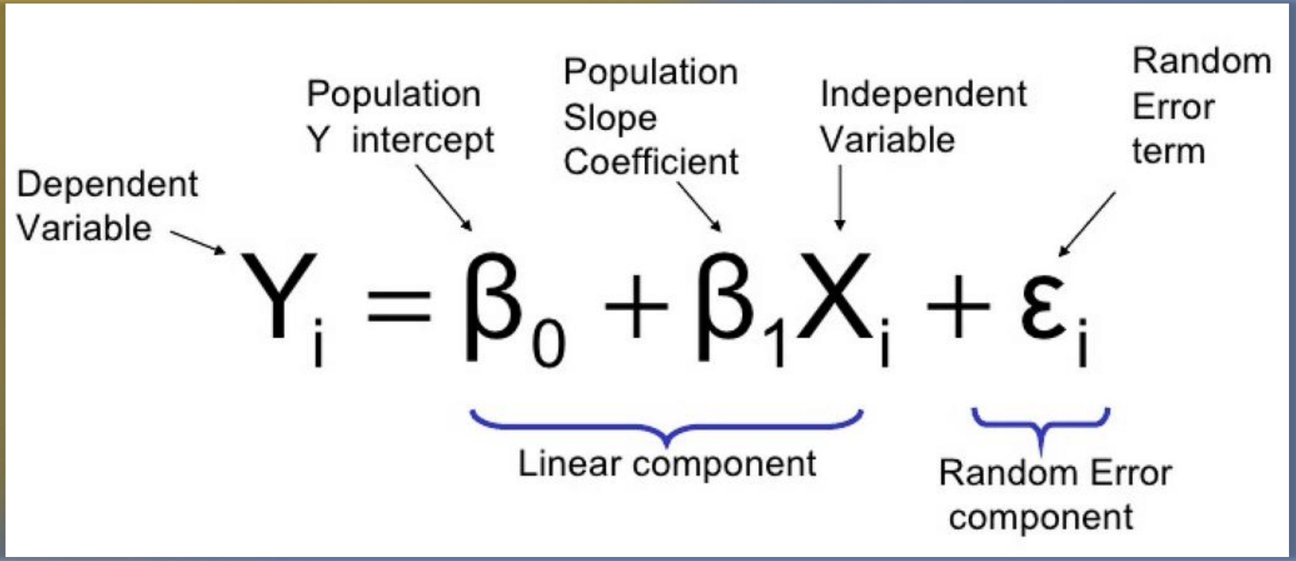
Mikor használunk regressziót?

- Tipikusan akkor, amikor meg akarjuk mutatni, **hogy bizonyos jelenségek hogyan befolyásolnak másokat**, ha egyáltalán befolyásolják, vagy bizonyos változók milyen kapcsolati rendszer szerint hozhatók összefüggésbe.
- A regressziós eljárások akkor is hasznosnak bizonyulnak, amikor valamilyen **választ szeretnénk előre jelezni**.
- Például: meg lehet jósolni egy családi ház energia felhasználását a következő órára, ha tudjuk hogy milyen az időjárás, a napnak melyik órájában járunk, mekkora a ház, hányan lakják.
- A fenti példában melyik a célváltozó és melyik a prediktor? És a képen?



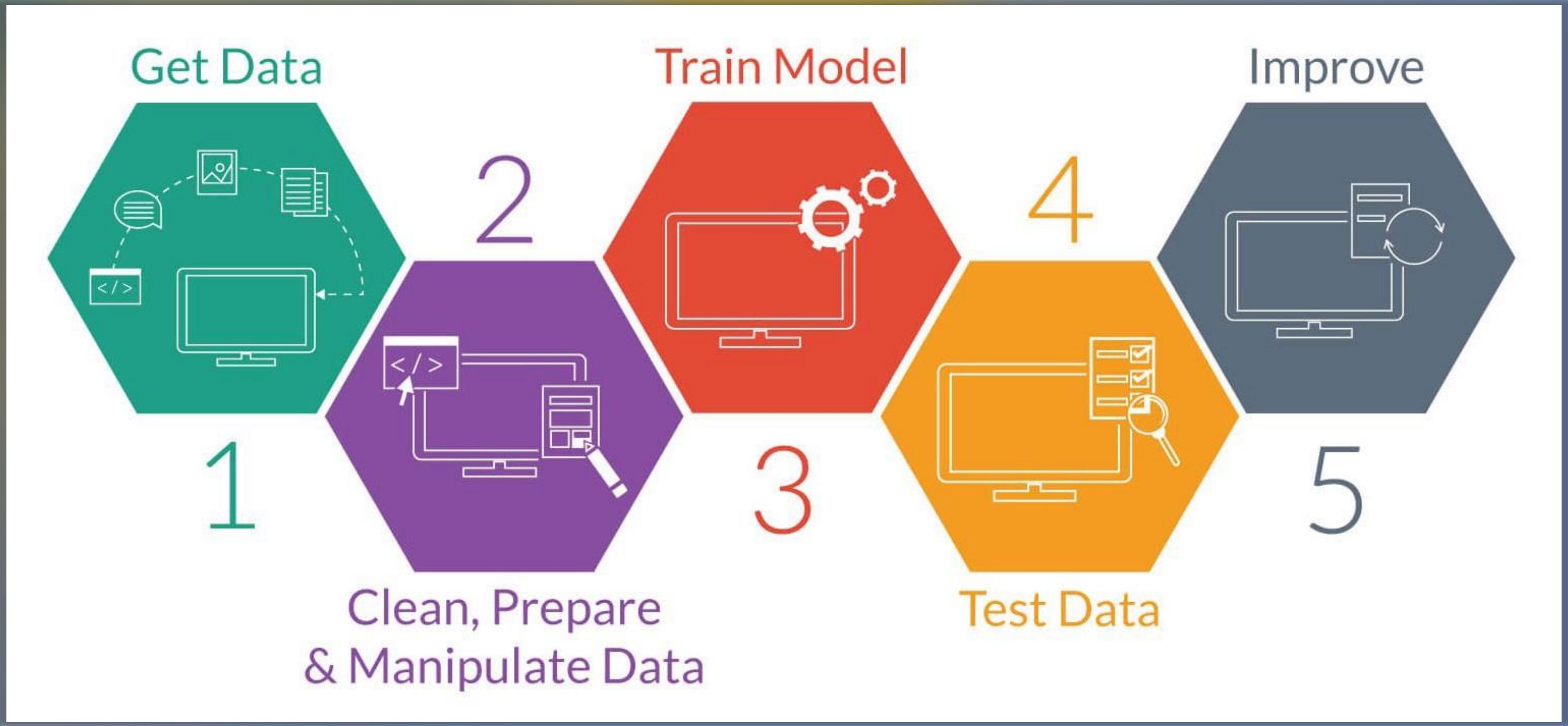
Lineáris regresszió: a probléma felírása

- A lineáris regresszió esetén valamely y változót szertnénk megbecsülni valamely $x = (x_1, \dots, x_r)$ adathalmaz alapján.
- A regressziós egyenletet formálisan (kép): A β értékek a **regressziós együtthatók**, ε pedig a véletlen hiba.
- A lineáris regresszió az együtthatókra ad becslést.
- A becsült regressziós függvényt a következőképp írjuk fel: $f(x) = b_0 + b_1 x_1 + \cdots + b_r x_r$
- Ahol $b_0 \dots b_r$ a **becsült regressziós paraméterek**, másnéven **súlyok**.



A regresszió teljesítménye

- A teljesítményt azért kell mérni, mert szeretnénk tudni, milyen jó a modell. Ehhez definiálni kell, mi az, hogy „jó”?
- A becsült outputoknak olyan közel kell esnie a valós értékekhez, amennyire csak lehetséges. Ez nem mindig jelent majd 100% egybeesést.
- A valós és becsült értékek különbsége a **rezidum.**
- A rezidumok összege a **hiba.**
- A regresszió művelete a legkisebb rezidumokhoz tartozó paraméterek megkeresése.



Rezidumok kiszámítása SSR módszerrel

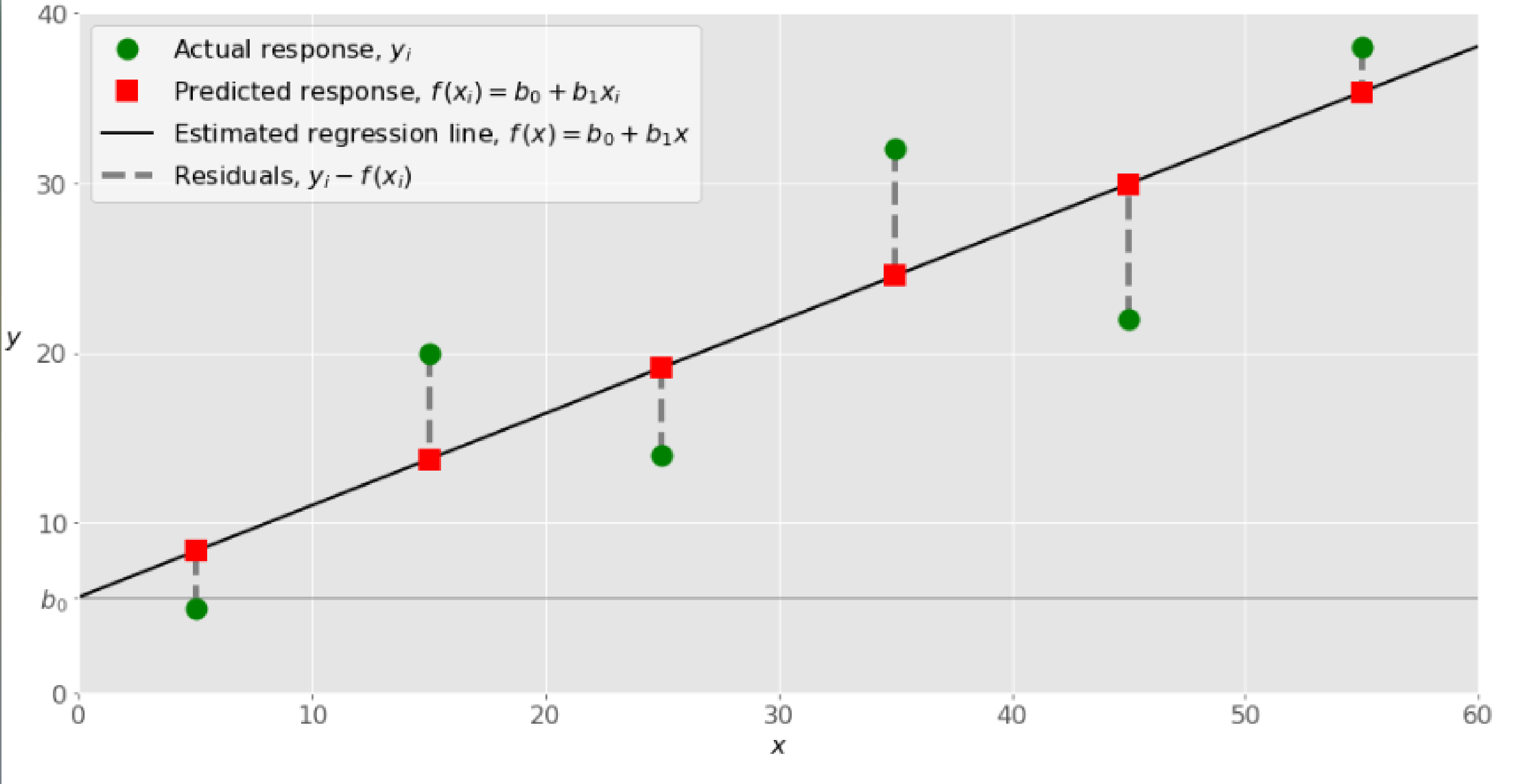
- A rezidum valamely függvény szerint áll elő, amelynek inputja a valós és a becsült érték. Ez a **költségfüggvény.** Az eljárás célja ennek a **minimalizálása** olyan módon, hogy a valós kapcsolatokat hitelesen leképezze.
- Az egyik leggyakoribb ilyen a négyzetes hiba:

$$SSR = \sum (y_i - f(x)_i)^2$$

(Sum of Squared Residuals)

- A költségfüggvény általános alakja: $L(y_i, f(x))$

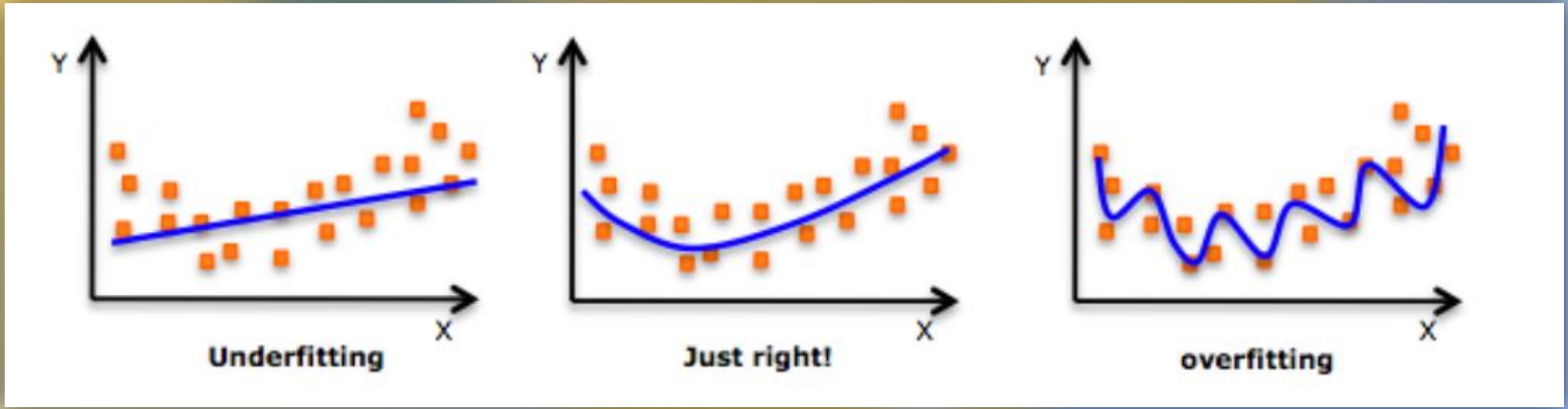
(L: Loss function)



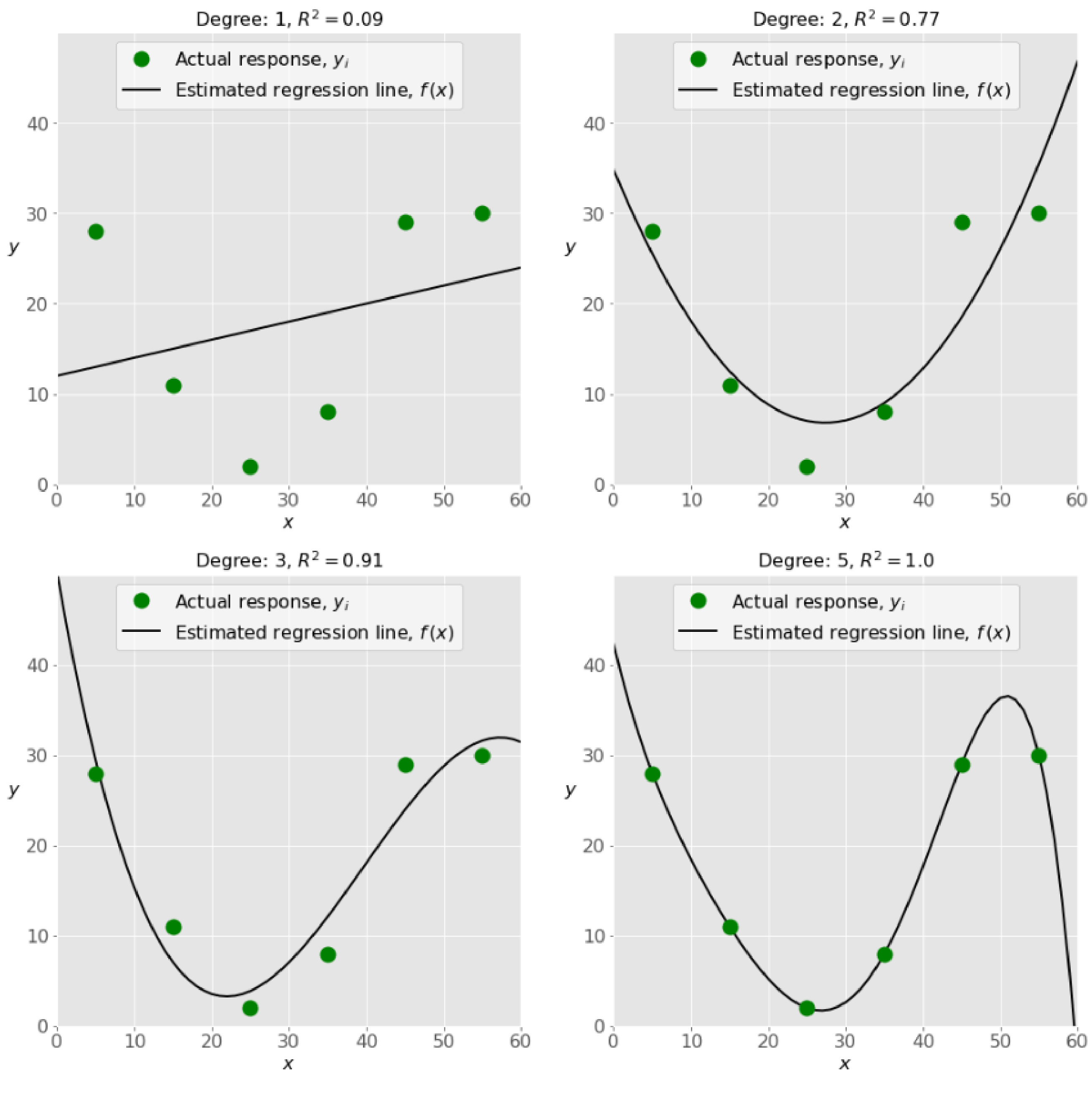
Example of simple linear regression

Alultanulás és túltanulás

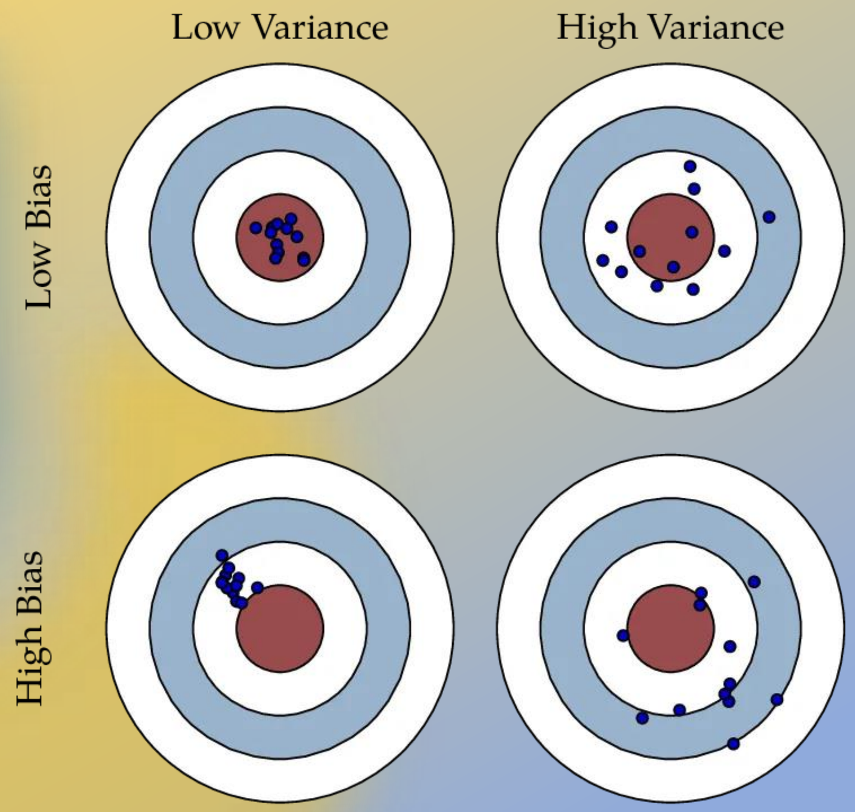
- Az csak az egyik feladat, hogy a függvény illesztése pontos legyen.
- A másik viszont, hogy a létrehozott modell ne torzuljon el olyan mértékben, hogy az a valóság kapcsolatait meghamisítsa.



Soroljuk be az alábbi modelleket: alultanult, pontos vagy túltanult?



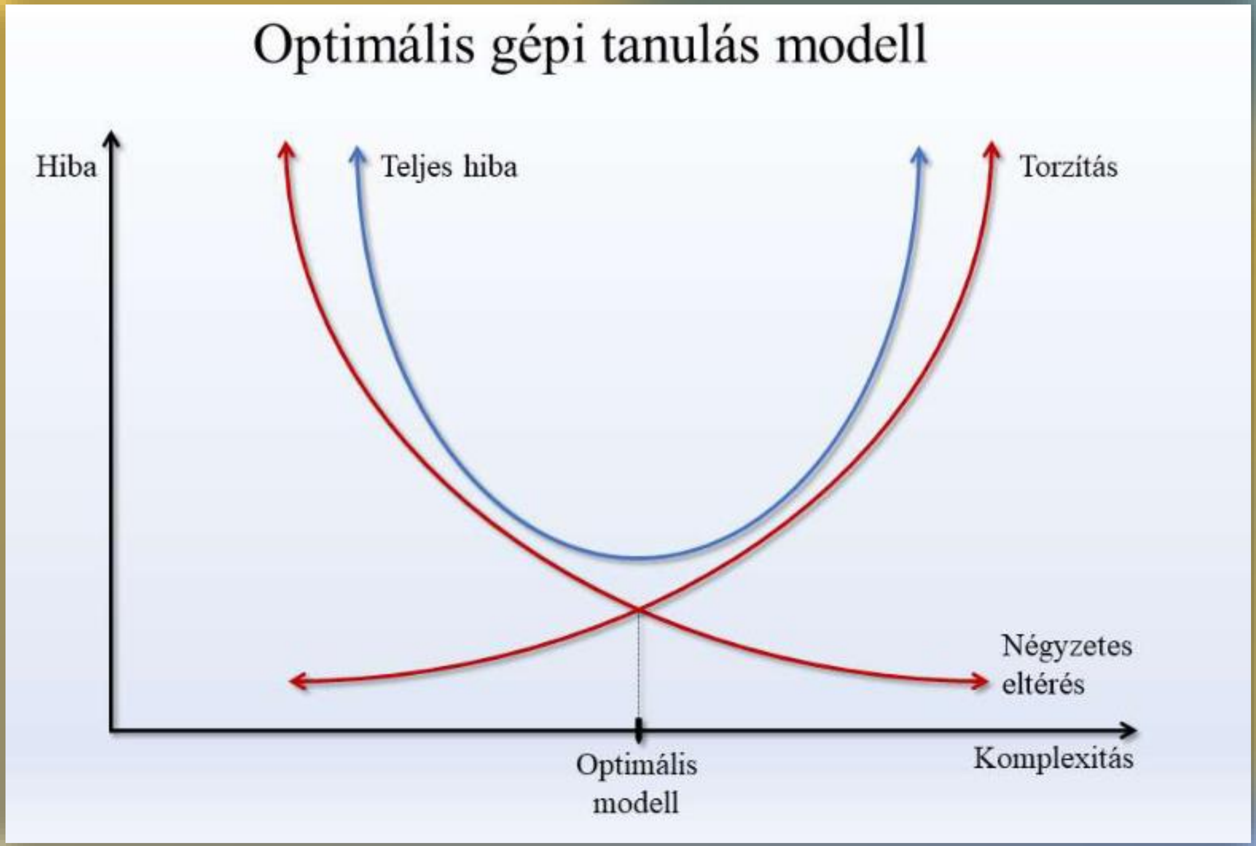
Bias/Variance tradeoff

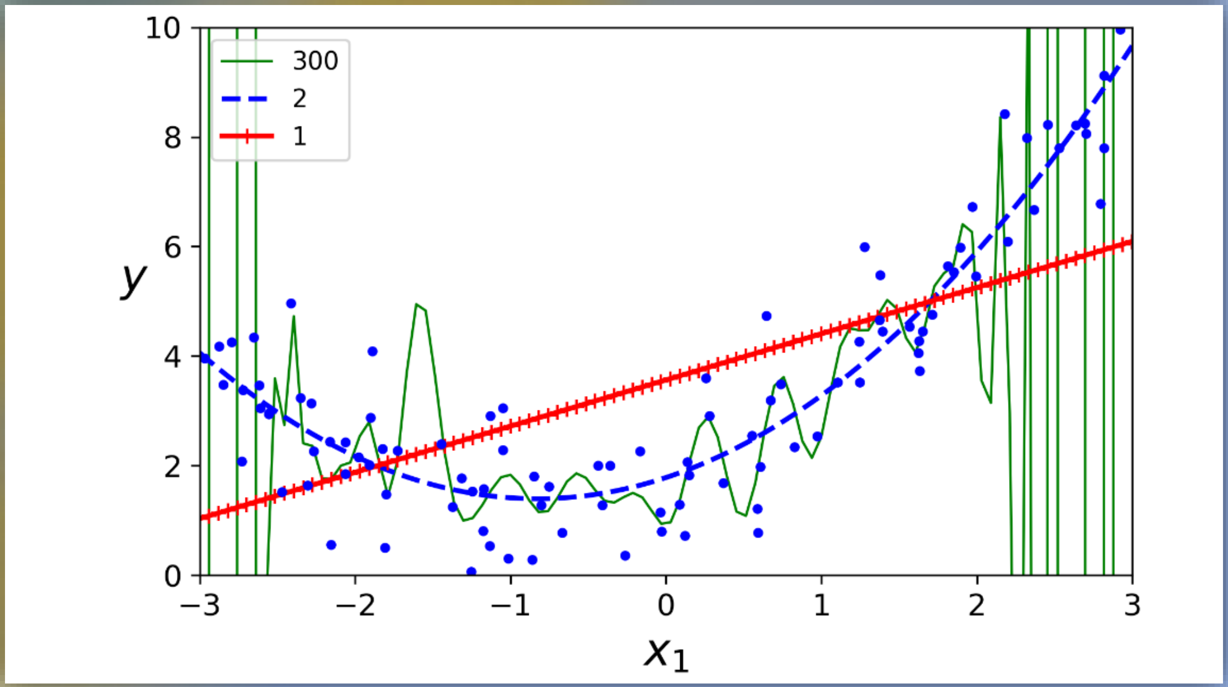


- Az optimális modell két végpontja a torzítás és a variancia.
- **Torzítás** (Bias): A generalizációs hibának ezen része a helytelen feltételezésekből ered. Pl.: Az a feltételezés, hogy a változók között lineáris kapcsolat található, miközben a valóságban kvadratikus.
- **Variancia** (Variance): A modell tanító adathalmazban lévő kisebb kilengések irányában mutatott érzékenysége. Egy magas szabadságfokú modellnek valószínűsíthetően a varianciája is magas lesz. Mit eredményez a túlságosan magas szabadságfok?

A feladat tárgya tehát az **optimális** modell!

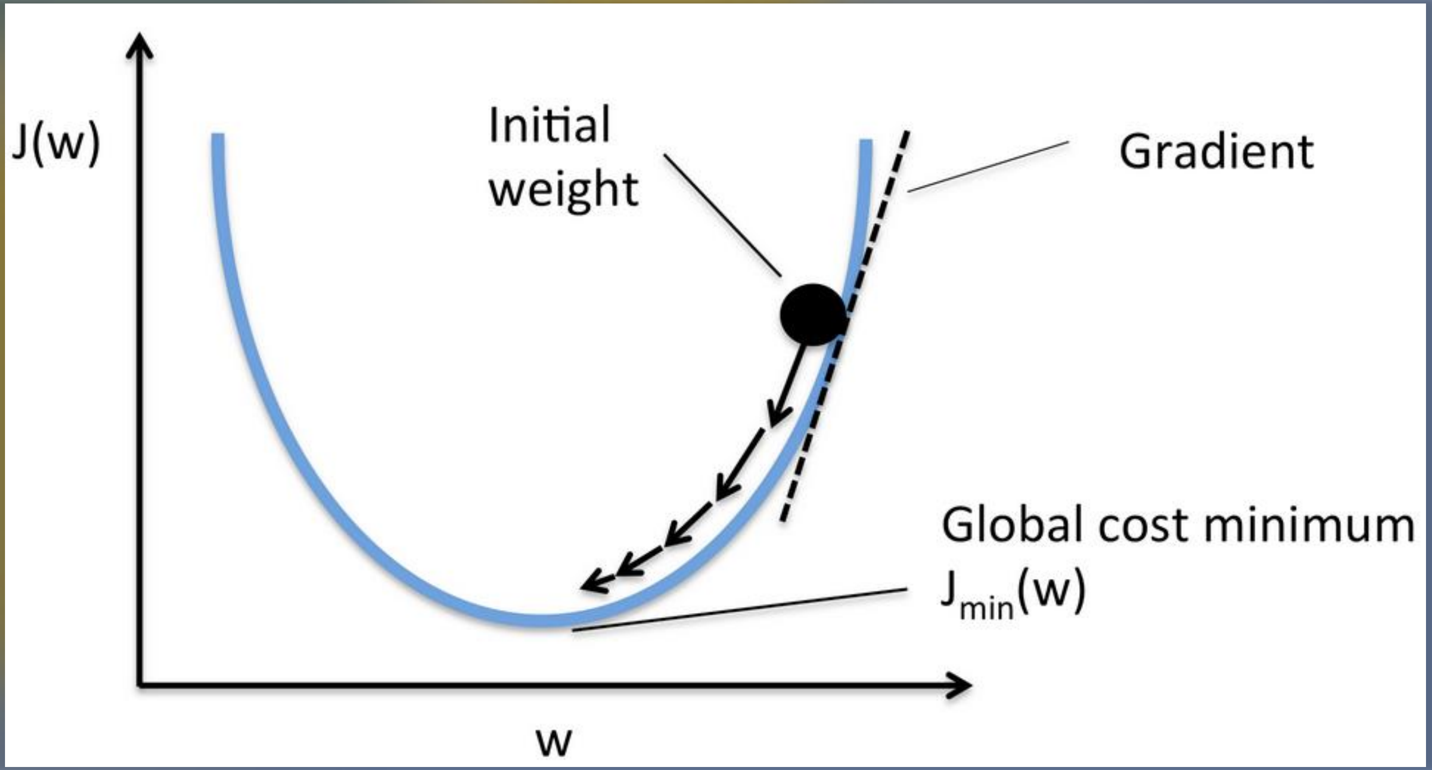
- Modellezés során figyelembe kell venni, hogy a modell pontosan illeszkedjen az adatokra.
- Viszont azt is, hogy a valóság kapcsolatait ne hamisítsa meg!





Optimalizálási módszer: **Gradiens Ereszkedés**

- A gradiens ereszkedés egy megoldást nyújt valamely célfüggvény minimalizálásának problémájára azáltal, hogy a paraméterek értékeit a célfüggvény gradienseivel ellentétes irányban frissíti.
- **A tanulási sebesség** (α vagy η -nű) az egyes paraméter frissítések nagyságát adja meg.
- Nagy tanulási sebesség: gyors tanulás (és fordítva).
- Más szóval: a költségfüggvény által generált lejtőt lefelé követjük, ameddig el nem érünk egy völgyet. Az, hogy mekkorát lépünk egyszerre, a tanulási sebességen múlik.



Gradiens ereszkedés eljárásai

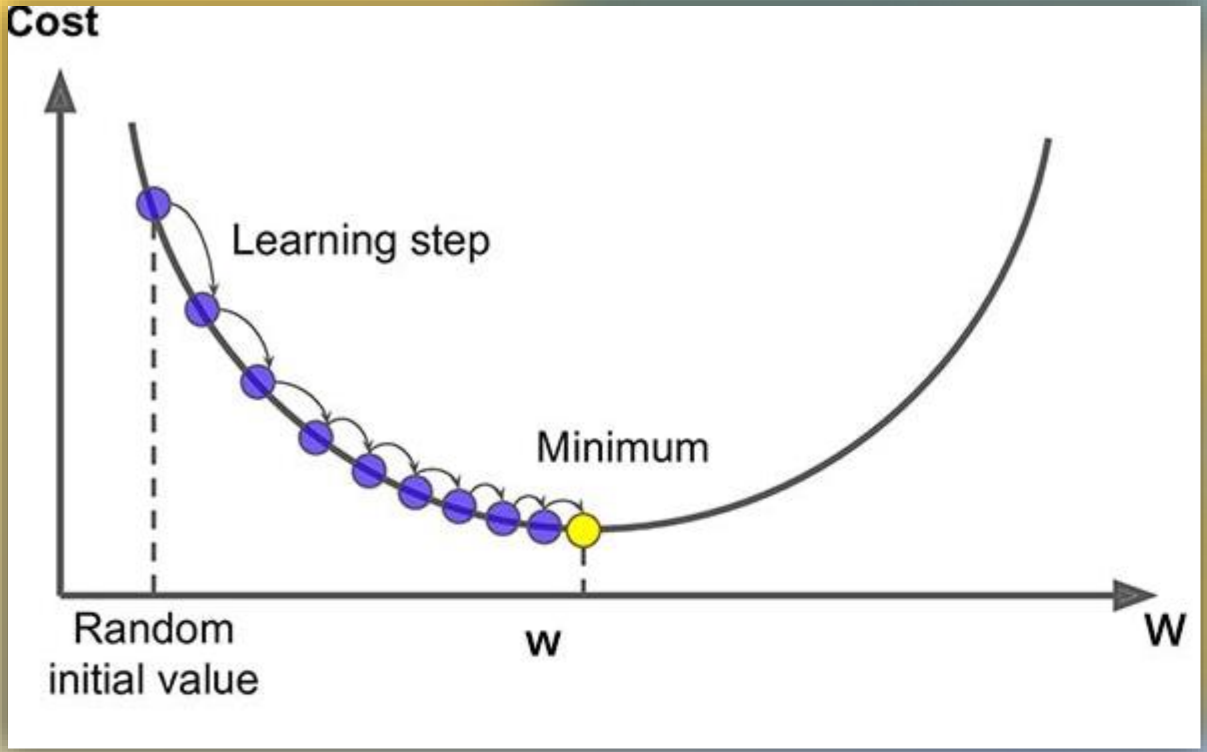
- Több eljárásmód is létezik: AdaGrad, Adadelta, Nadam, RmsProp…
- Az „alap” algoritmus gradiens ereszkedésre:

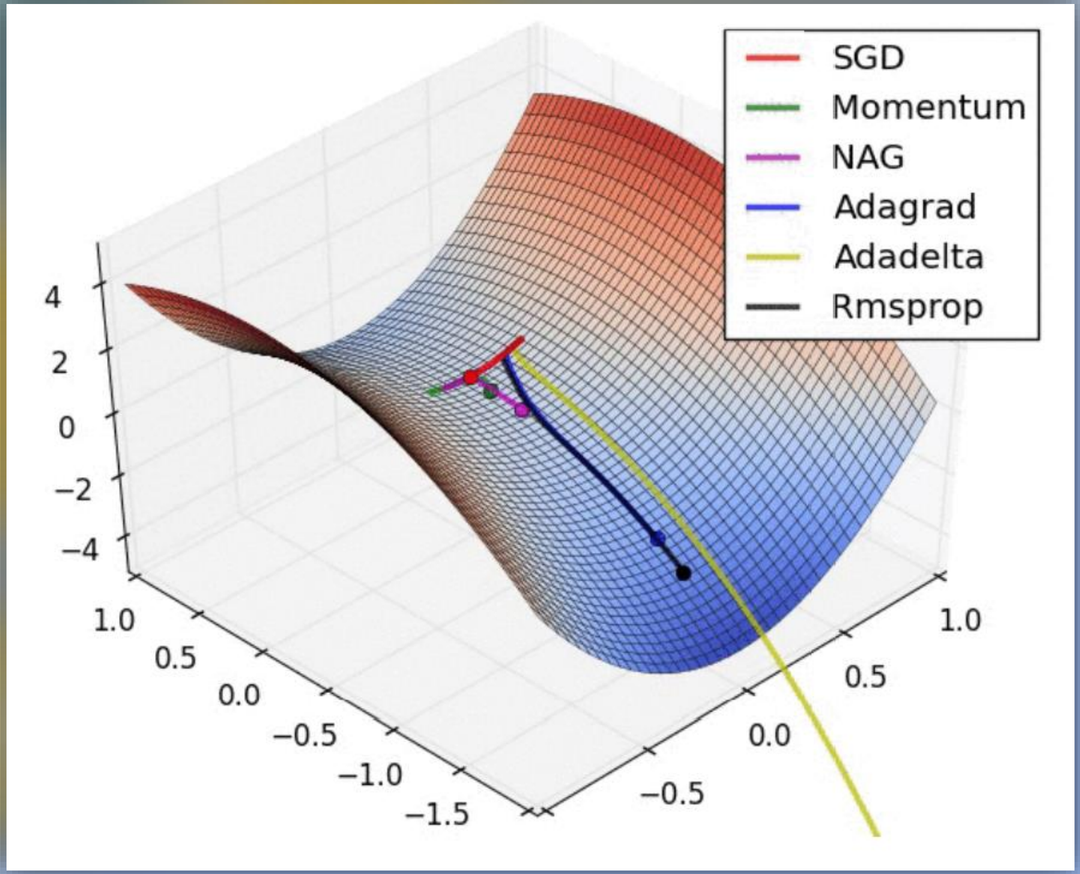
```
for i in range(nb_epochs):
  params_grad = evaluate_gradient(loss_function, data, params)
  params = params - learning_rate * params_grad
```

- Képlettel felírva: $$\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$$
  - θ: paraméterek vektora (théta)
  - η: tanulási sebesség (nű)
  - ∇_θ: paraméter gradiense (nabla vagy del)
  - $J(\theta)$: a költség nagysága adott paraméter vektor szerint
- Ez adja meg, lépésenként mennyit változzon egy paraméter. Pl. meredekség.

Gradiens ereszkedés a minimum irányába

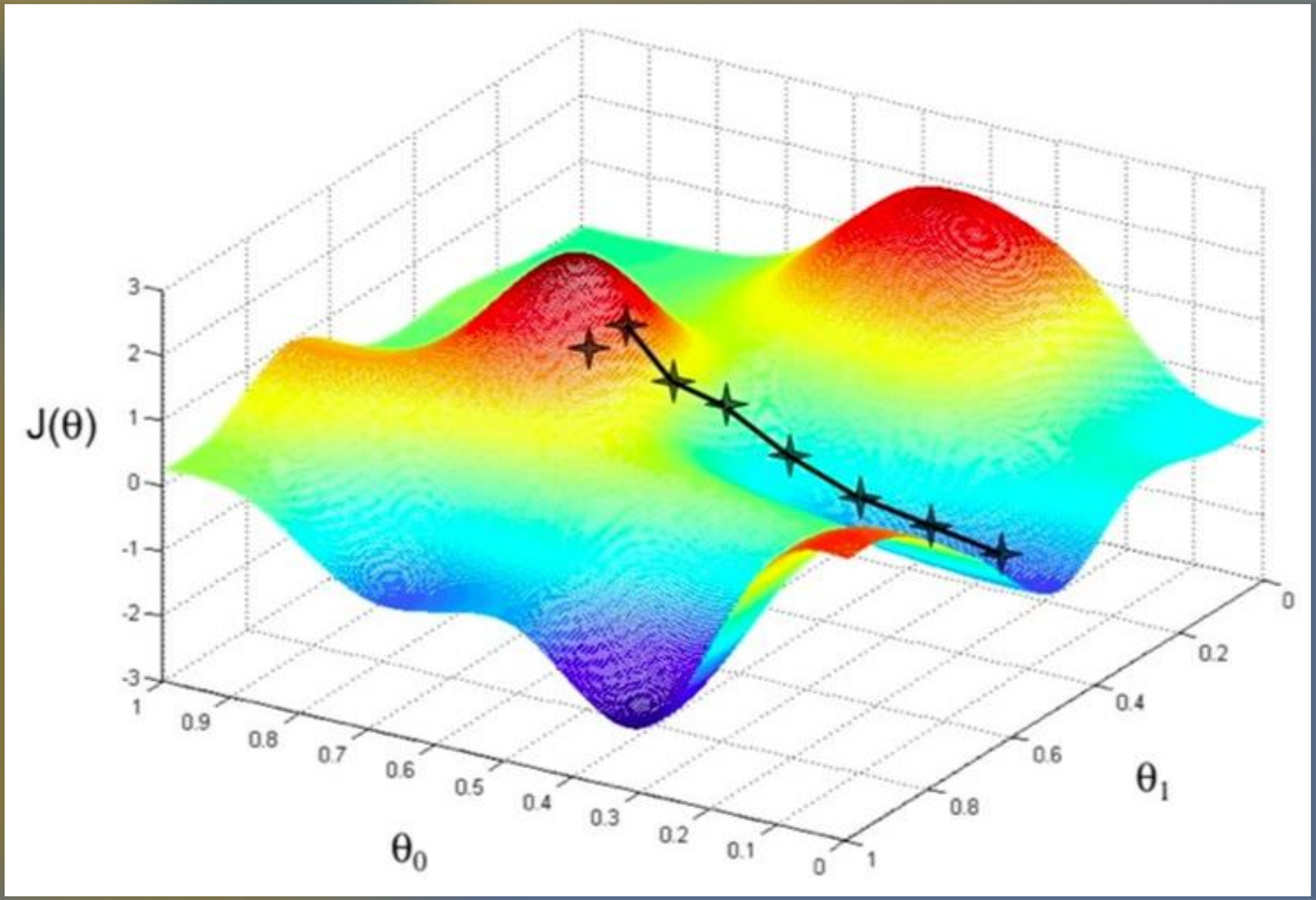
- Az eljárás nem garantáltan ér el globális minimumot!
- Beragadhat egy lokális extrém helyen.
- Ez konvex függvényeknél nem probléma.





Gradiens ereszkedés több dimenzióban

- A paraméterek által alkotott tér két paraméter esetén.
- $J(\theta)$: a költség.
- θ_0: az első paraméter: eltolás, vagy az y tengely metszéspontja.
- θ_1: a második paraméter: meredekség.
- Ezután a polinomikus paraméterek következnek.



Házi feladat

- https://www.youtube.com/watch?v=sDv4f4s2SB8