Regresija

Jure Žabkar
jure.zabkar@fri.uni-lj.si

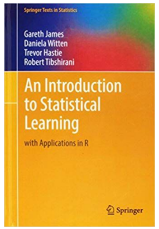
4. 5. 2021

A.I.LAB
Ljubljana

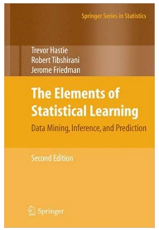
Vsebina

- Regresija
- Regularizacija
- Metode za izbiro atributov
- k-NN regresija
- Regresijska drevesa

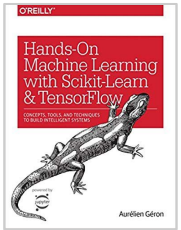
Literatura



3.1 - 3.3



2.3.1, 3.2

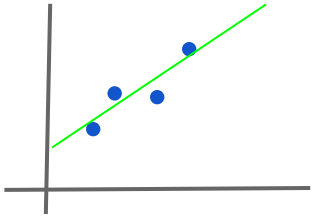


str. 102-130

Strojno učenje

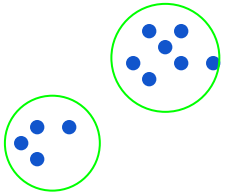
Nadzorovano

Regresija, klasifikacija

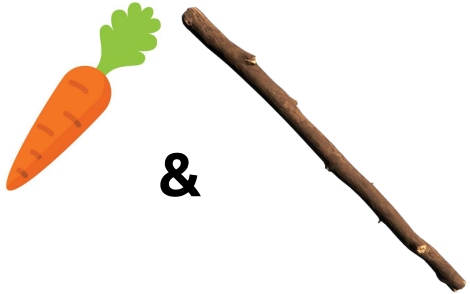


Nenadzorovano

Gručenje, povezovalna pravila



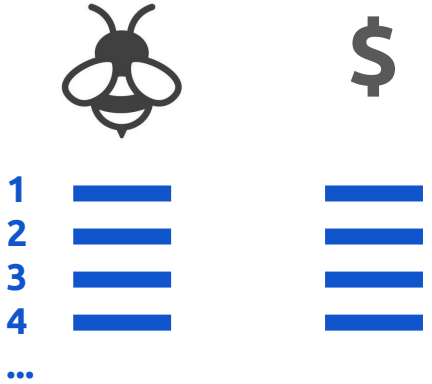
Spodbujevano učenje



Linearna regresija

Pridelek medu

$ = A ×  + B



Regresija

Množica podatkov **D**:

{**e** = [x_1, … x_n, **y**] | x_i **vrednosti atributov**,

y vrednost razreda}

Vrednosti **atributov**:

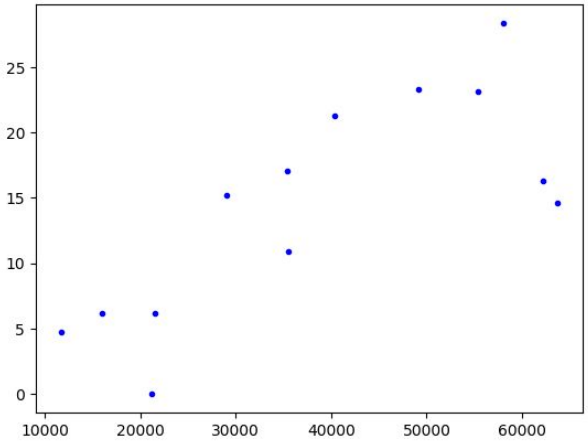
lahko **diskretne** ali **zvezne** vrednosti

Vrednost **razreda**: **zvezna**

Linearna regresija (hitra ponovitev)

Univariatna $$y_i = \beta_0 + \beta_1 x_i$$

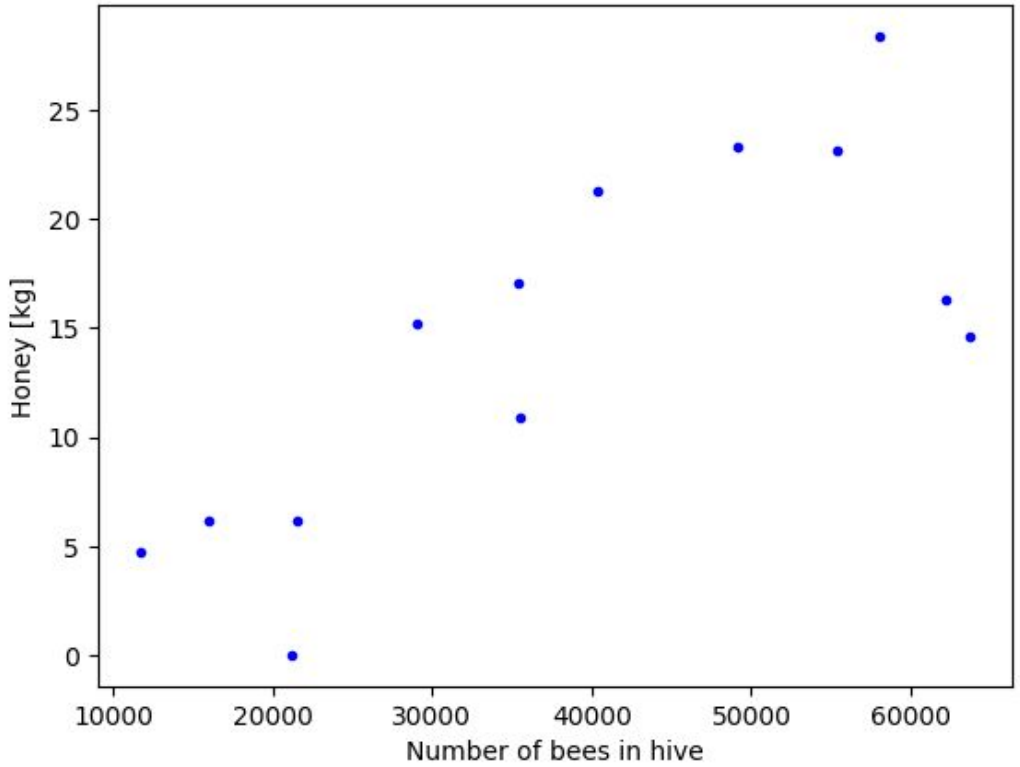
Multivariatna $$Y = X\beta + \epsilon$$



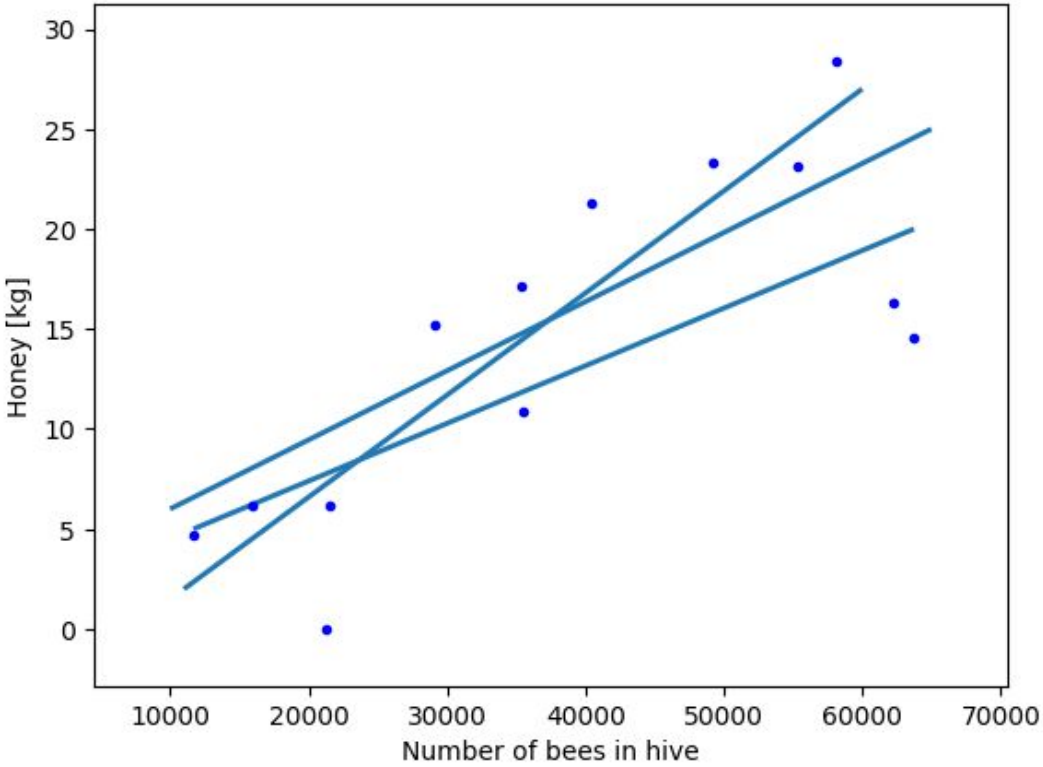
Analitična rešitev

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

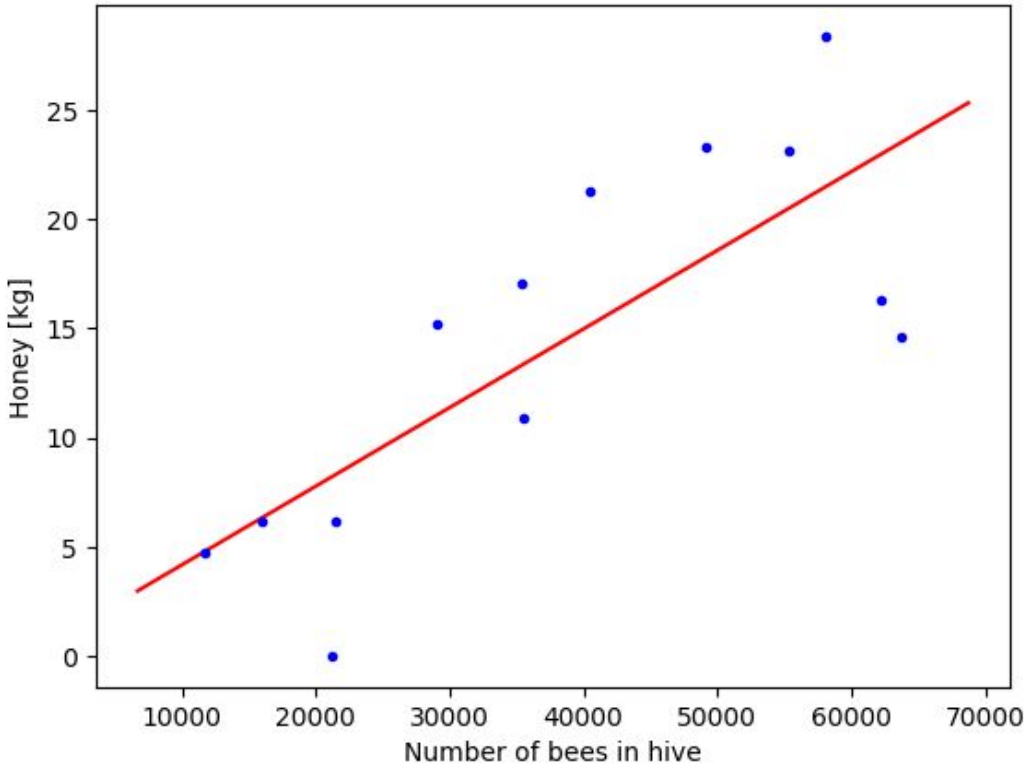
Pridelek medu na panj



Pridelek medu na panj, **možni modeli**



Pridelek medu na panj, najboljši model



Med = f (#čebel, Temperatura, Vlaga)

$$Y = X\beta + \epsilon$$

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Izračun parametrov

$$Y = X\beta + \epsilon$$

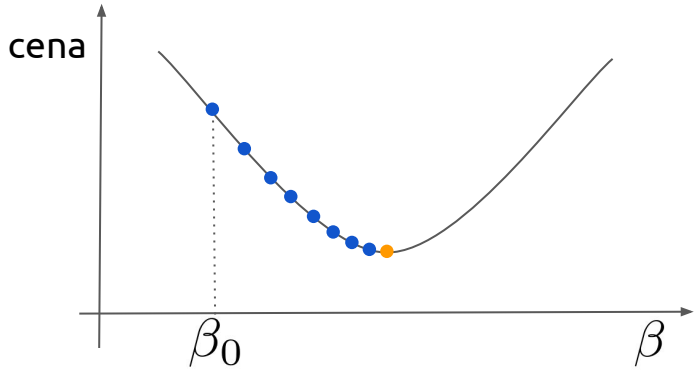
$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Računanje inverza problematično:

- če **$X^T X$ singularna**, npr. če je v podatkih več atributov (n), kot primerov (m), torej **m < n**, ali če so v podatkih odvečni atributi.
- **računska kompleksnost** zelo velika, $O(n^3)$
- **preveč podatkov**, da bi jih lahko shranili v spomin računalnika.

Deloma je problem rešljiv, če namesto inverza računamo psevdoinverz.
(tako deluje npr. LinearRegression() v Scikit-Learn)

Gradientni spust



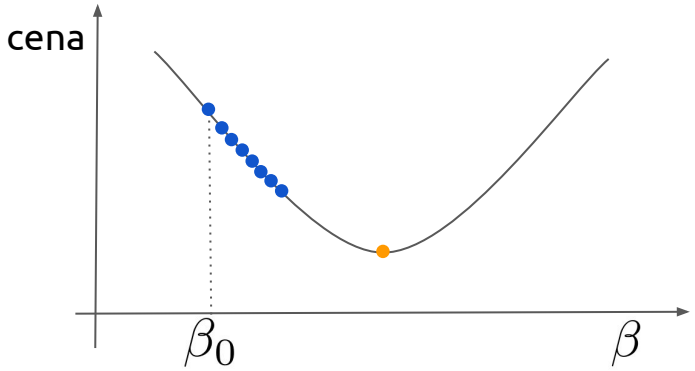
- Splošen optimizacijski algoritem
- Ideja:
  iterativno spreminjamo parametre in minimiziramo cenovno funkcijo
- Parametre inicializiramo z naključnimi vrednostmi
- Cenovna funkcija:

RMSE = $\sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$ (običajno kar MSE)

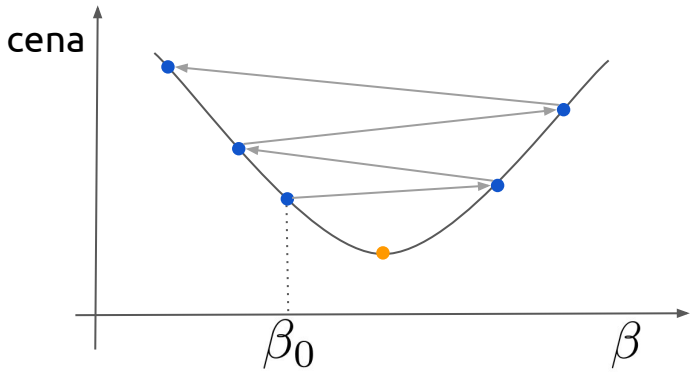
Gradientni spust

Parameter **velikost koraka**

- če premajhna, počasna konvergenca algoritma



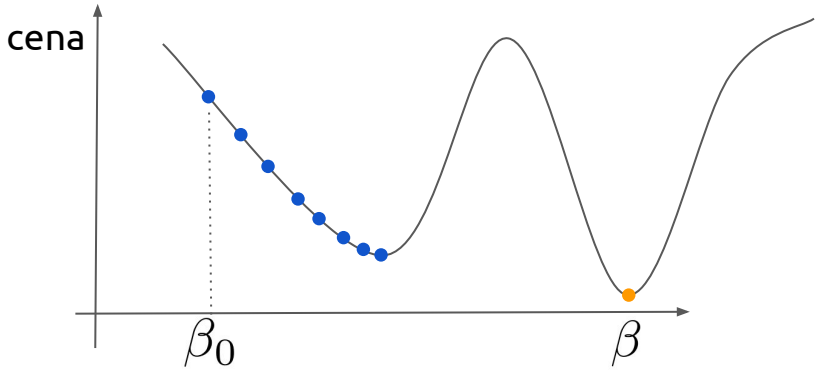
- prevelika lahko povzroči divergenco



Gradientni spust *(angl. Gradient Descent)*

V splošnem se lahko ujame v **lokalni minimum.**

Ker je MSE konveksna funkcija, je lokalni ekstrem tudi **globalni.**



Gradientni spust, implementacije

- Gradient spust v sklopu *(Batch Gradient Descent)*
- Naključni gradientni spust *(Stochastic Gradient Descent)*
- Gradientni spust v mini-sklopih *(Mini-batch Gradient Descent)*

Gradientni spust v sklopu *(Batch Gradient Descent)*

- Na celotni učni množici izračunamo vse gradiente naenkrat
- Zelo počasno na podatkih z veliko učnimi primeri
- Učinkovit pri velikem številu atributov v primerjavi z normalno enačbo

$$MSE = \frac{1}{m} \sum_{i=1}^{m} \left(\beta^T x^{(i)} - y^{(i)} \right)^2$$

$$\frac{\partial}{\partial \beta_j} MSE(\beta) = \frac{2}{m} \sum_{i=1}^{m} \left(\beta^T x^{(i)} - y^{(i)} \right) x_j^{(i)}$$

$$\nabla_\beta MSE(\beta) = \begin{bmatrix} \frac{\partial}{\partial \beta_0} MSE(\beta) \\ \vdots \\ \frac{\partial}{\partial \beta_m} MSE(\beta) \end{bmatrix} = \frac{2}{m} X^T (X\beta - y)$$

Naključni gradientni spust *(Stochastic Gradient Descent)*

- Rešuje problem počasnosti BGD
- Na vsakem koraku **naključno** izbere učni primer in izračuna gradiente v njem
- Naključnost koristi tudi v primeru nekonveksne cenovne funkcije (preprečuje ujetost v lokalne min.)
- Ustavitev preiskovanja: zmanjševanje velikosti koraka (simulirano ohlajanje)

Gradientni spust v mini-sklopih *(Mini-batch Gradient Descent)*

- Kombinacija **BGD** in **SGD**
- Na vsakem koraku naključno izbere podmnožico učnih primerov
- V primerjavi s SGD manj skače po prostoru parametrov; posledično težje uide lokalnemu ekstremu, če cenovna funkcija ni konveksna

Napovedna točnost regresijskih modelov

- RMSE = $\sqrt{\frac{\sum_{i=1}^{k} y^{(i)} - \hat{y}^{(i)}}{k}}$

Če hočemo, da nam ta številka kaj pove, moramo poznati enote razreda

- $R^2 = 1 - \frac{\sum_k (y^{(i)} - \hat{y}^{(i)})^2}{\sum_k (y^{(i)} - \bar{y})^2}$

Vsota kvadratov napak našega modela, recimo linearne regresije.

Vsota kvadratov napak, če bi model vedno napovedoval kar povprečno vrednost razreda iz učne množice.

Delež razložene variance:

- **Dober model** - števec ulomka gre proti 0, R^2 proti 1.
- **Slab model** - ulomek gre proti 1, R^2 proti 0.

Pristranskost in varianca

Napaka modela je sestavljena iz treh vrst napak:

1. **Pristranskost** (bias): napaka zaradi napačnih predpostavk (predpostavimo, da je odvisnost v podatkih linearna, a je v resnici kvadratna). Tak model se **premalo** prilega podatkom.

2. **Varianca** (variance): napaka zaradi občutljivosti modela na majhne spremembe v podatkih. Kompleksnejši modeli imajo večjo varianco in se lahko **pretirano** prilagodijo podatkom.

3. **Neodpravljiva napaka** (irreducible error), npr. šum v podatkih

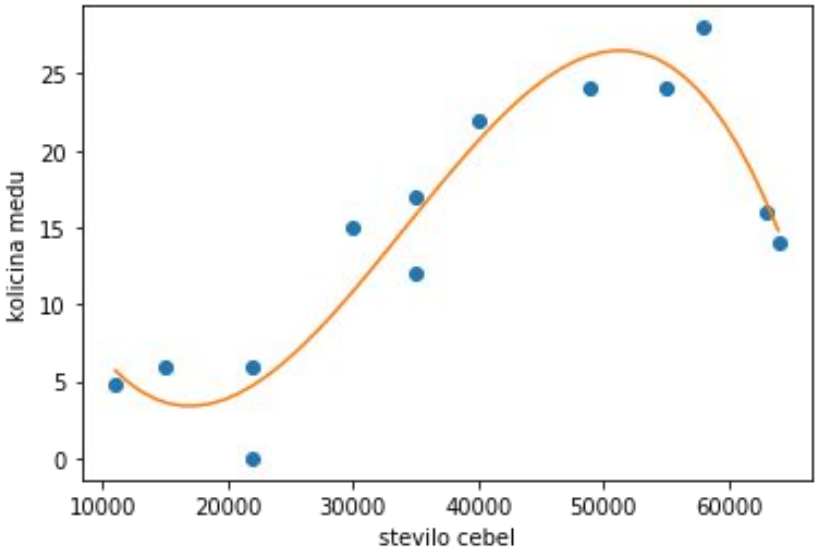
Polinomska regresija

Dodajmo v atributni prostor nove atribute tako, da obstoječe polinomsko razširimo:

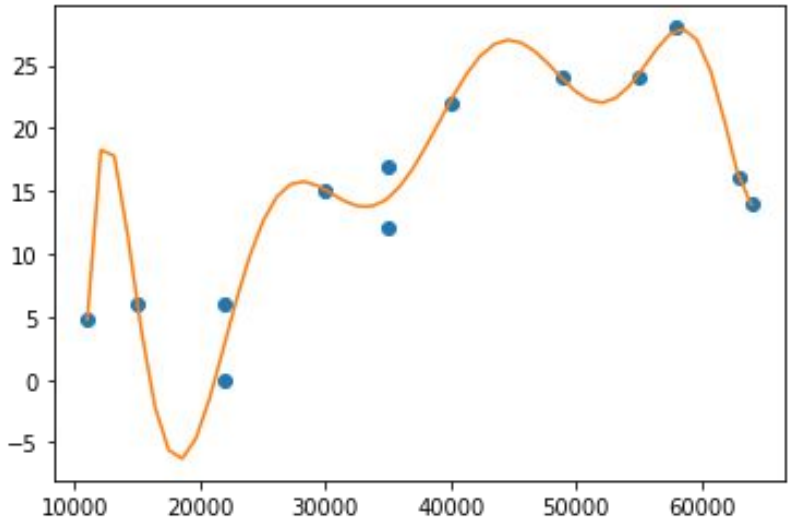
x … število čebel

Dodajmo še x^2 in x^3 in izračunajmo koeficiente regresijskega modela.

Polinomska regresija



[x0 x1 x2 x3]
[0. 16.53 -4.41 -6.05]



[x0 x1 x2 x3 x4 x5 x6 x7 x8 x9]
[0. 32.03 16.05 -133.72 -50.91 221.49 34.88 -131.92 -6.95 25.32]

Regularizacija

je preprečevanje pretiranega prilagajanja modela učnim podatkom.

Polinomi visoke stopnje se očitno preveč prilagodijo učnim podatkom.

Kako se odraža to pretirano prilagajanje?
Absolutna vrednost koeficientov podivja.

Regularizacija

Nekako je treba ukrotiti koeficiente.
Cenovni funkciji dodamo člen, ki bo koeficiente po absolutni vrednosti minimiziral, npr.:

$$J(\beta) = MSE(\beta) + \alpha \sum_{i=1}^{m} \beta_i^2$$

S parametrom alfa uravnavamo stopnjo regularizacije (tipično alpha=0.1 ali manj).

Regularizacija linearne regresije

- Ridge

$$J(\beta) = MSE(\beta) + \alpha \sum_{i=1}^{n} \beta_i^2$$

- Lasso

$$J(\beta) = MSE(\beta) + \alpha \sum_{i=1}^{n} |\beta_i|$$

Ni odvedljiva!
Ni težava, če uporabljamo gradientni spust.

- Elastic Net

$$J(\beta) = MSE(\beta) + r\alpha \sum_{i=1}^{n} |\beta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^{n} \beta_i^2$$

Regularizacija, navodila za uporabo

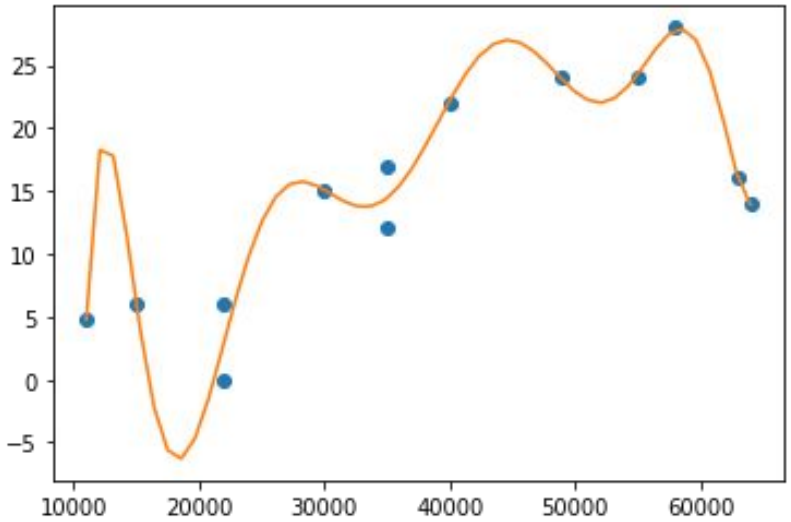
linearna regresija brez regularizacije: **praktično ni priporočljivo**

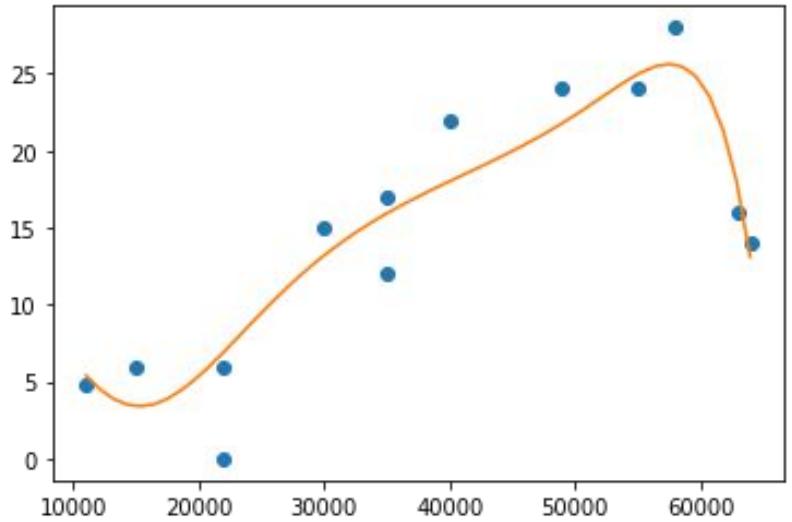
Ridge: dobra privzeta regularizacija

Če sklepamo, da so nekateri atributi odveč, uporabimo Lasso ali Elastic Net, ker odstranita neuporabne atribute.

Elastic Net bolj priporočljivo od Lasso v primeru korelacije atributov ali ko je atributov več od učnih primerov.

Ridge regresija

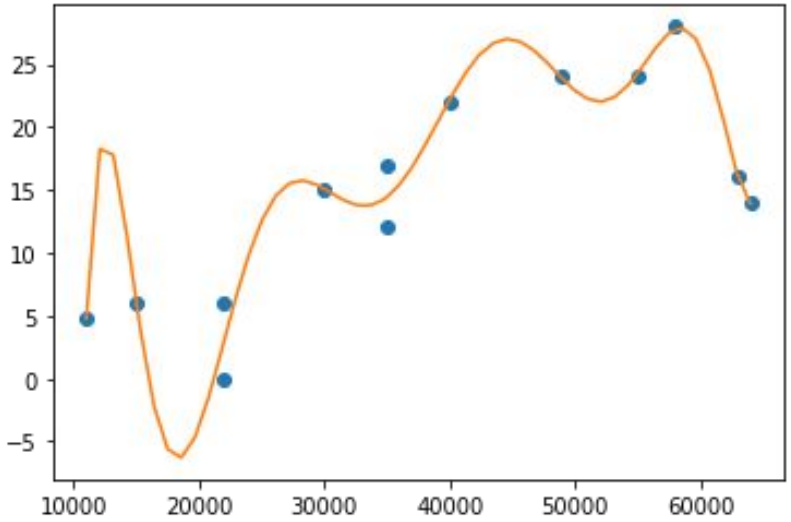


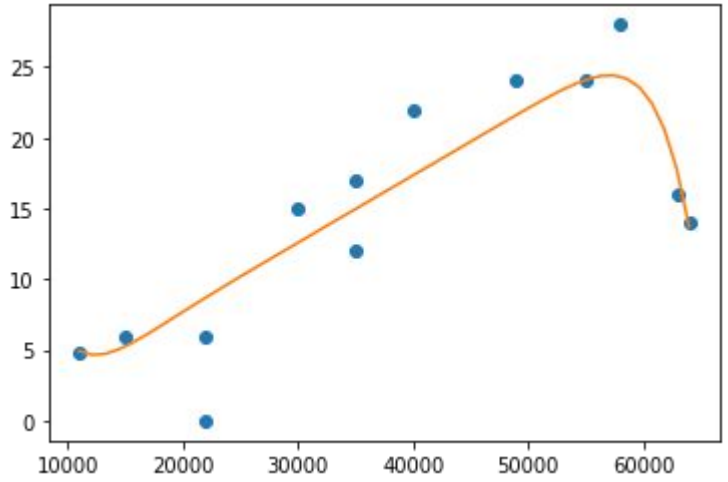


x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
[0.	32.03	16.05	-133.72	-50.91	221.49	34.88	-131.92	-6.95	25.32]

x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
[0.	6.99	-1.76	3.6	-0.29	0.64	0.97	-1.77	-0.62	0.13]

Lasso regresija

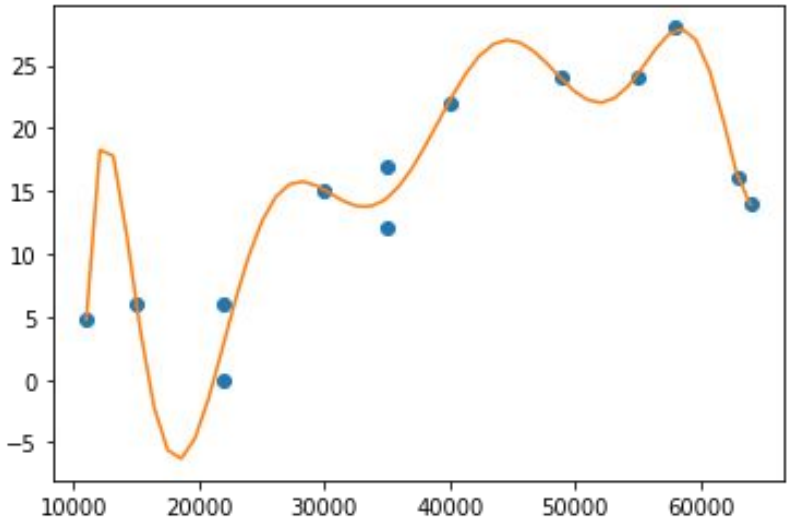


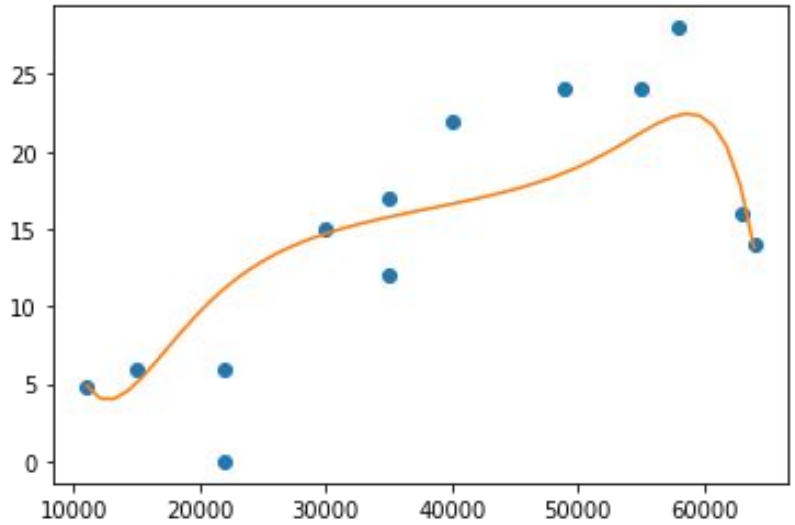


x0 x1 x2 x3 x4 x5 x6 x7 x8 x9
[0. 32.03 16.05 -133.72 -50.91 221.49 34.88 -131.92 -6.95 25.32]

x0 x1 x2 x3 x4 x5 x6 x7 x8 x9
[0. 8.3 0. 0. 0. 0. 0. 0. -0.34 -0.24]

ElasticNet regresija





x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
[0.	32.03	16.05	-133.72	-50.91	221.49	34.88	-131.92	-6.95	25.32]

x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
[0.	2.9	0.	1.98	0.	1.13	0.	0.	-0.35	-0.42]

Izbira atributov (feature selection)

Metode Filter *izbirajo atribute neodvisno od algoritma učenja; atribute filtrirajo preden začnemo z učenjem.*

Večinoma gre za računsko učinkovito statistično računanje, univariatno ali multivariatno, npr. ANOVA, chi^2, korelacije.

Izbira atributov (feature selection)

Metode Wrapper *z uporabo algoritmov učenja ocenjujejo podmnožice atributov in jih primerjajo med seboj. V grobem ločimo tri iterativne načine gradnje podmnožic atributov:*

- *naprej (začne z enim atributom in iterativno dodaja v model nove),*
- *nazaj (začne z vsemi atributi in jih iterativno izloča iz modela),*
- *koračno (dvosmerno iskanje) (step-wise, bi-directional). Hkrati naprej in nazaj; atributov, ki jih izberemo v koraku naprej, ne odstranimo v koraku od zadaj in obratno: tistih, ki jih odstranimo od zadaj, ne vključujemo v koraku naprej.*

Odkrijejo interakcije med atributi in poiščejo optimalno podmnožico za želeni algoritem učenja.

Izbira naprej (forward selection)

Iščemo najboljšo podmnožico množice p atributov.

1. Naj bo M_0 model **brez** atributov.
2. Za k = 0, …, p-1:
   a. Obravnavaj *p-k* modelov tako, da M_k **dodaš** 1 atribut
   b. Izberi najboljšega od teh *p-k* modelov (=M_{k+1});
      npr. tistega z največjim R^2
3. Izberi najboljši model izmed M_0, …, M_p z uporabo prečnega preverjanja in izbranega kriterija (napovedne točnosti, R^2, ...)

Izbira nazaj (backward selection)

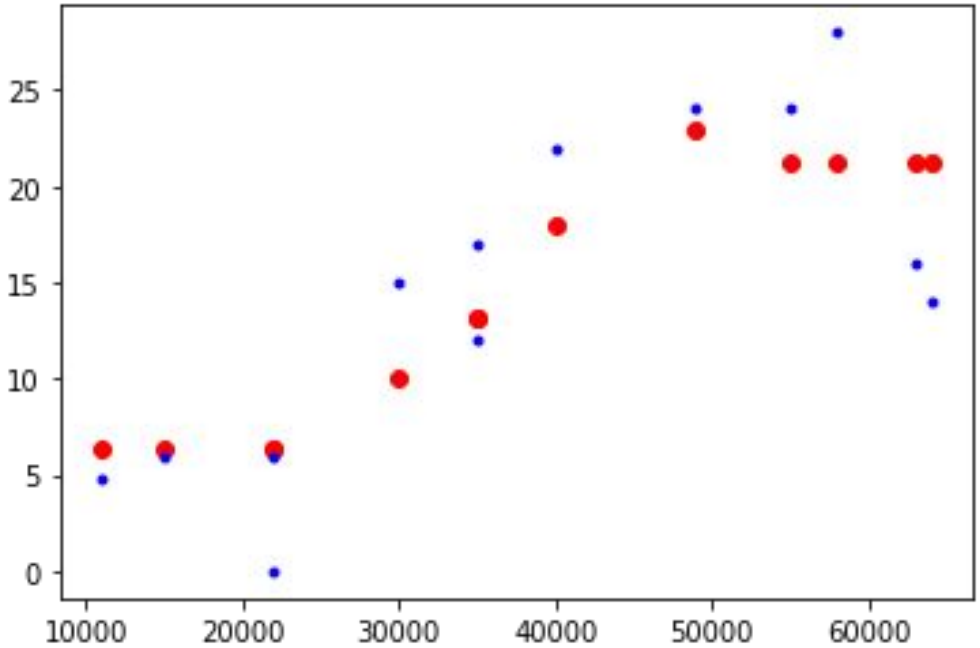
Iščemo najboljšo podmnožico množice p atributov.

1. Naj bo M_p model, ki vsebuje **vse** atribute.
2. Za k = p, …, 1:
   a. Obravnavaj *k* modelov tako, da iz M_k **izločiš** 1 atribut
   b. Izberi najboljšega od teh *p-k* modelov (=M_{k-1});
      npr. tistega z največjim R^2
3. Izberi najboljši model izmed M_0, …, M_p z uporabo prečnega preverjanja in izbranega kriterija (napovedne točnosti, R^2, ...)

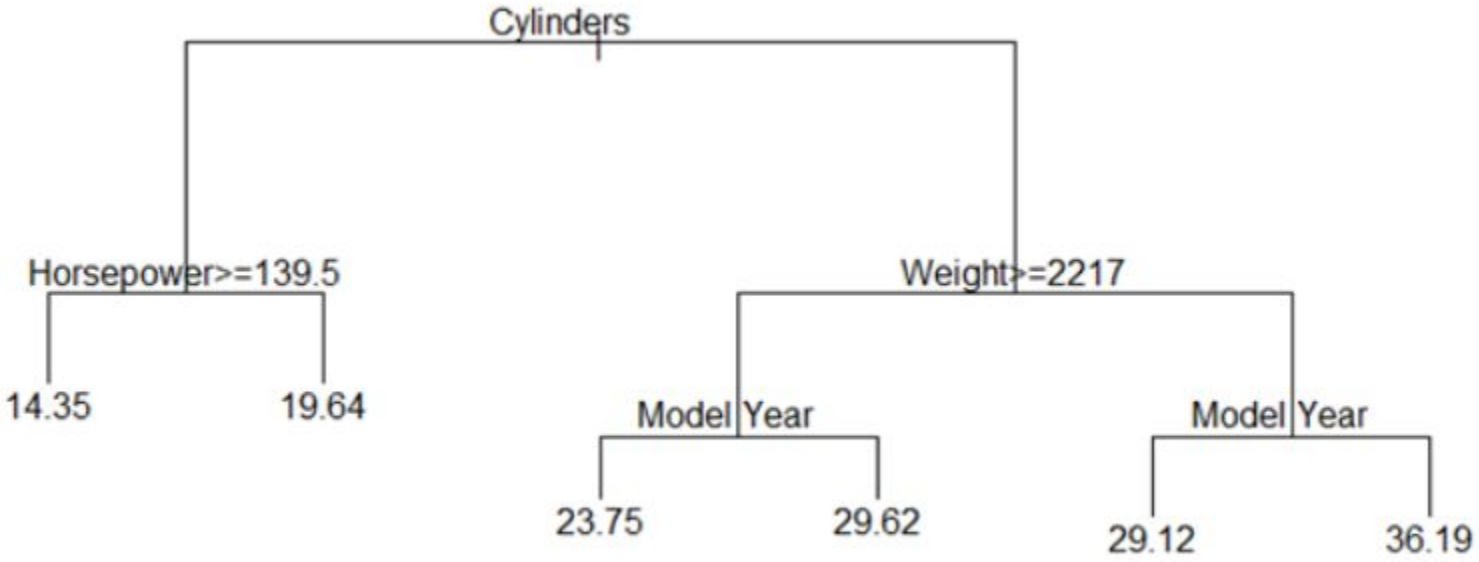
Izbira atributov (feature selection)

***Vgrajene** (Embedded) **metode** za izbiro atributov so del algoritmov strojnega učenja. Tipičen primer je algoritem za učenje odločitvenih dreves na podlagi informacijskega prispevka.*
Drugi primeri: Lasso, Elastic Net.

k-NN regresija



Regresijska drevesa



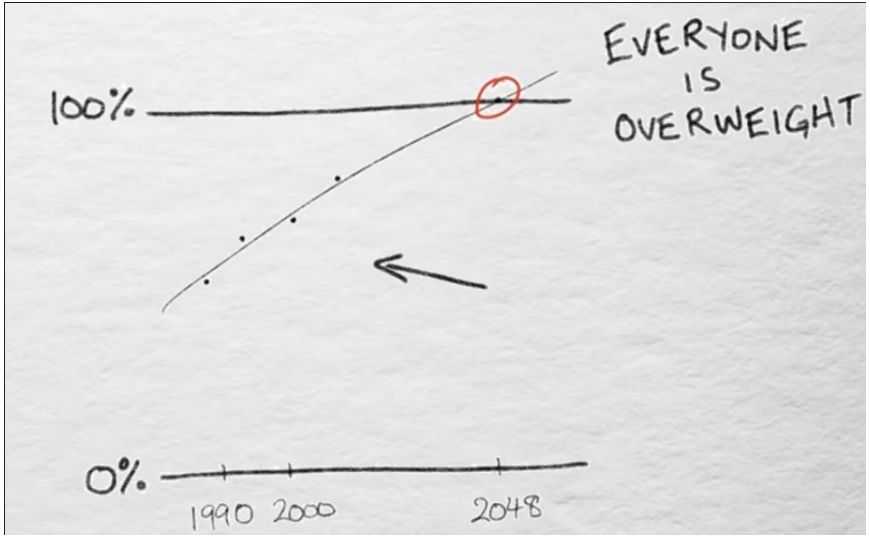
Bora Beran, On Anything Data, domena MPG

Skaliranje atributov

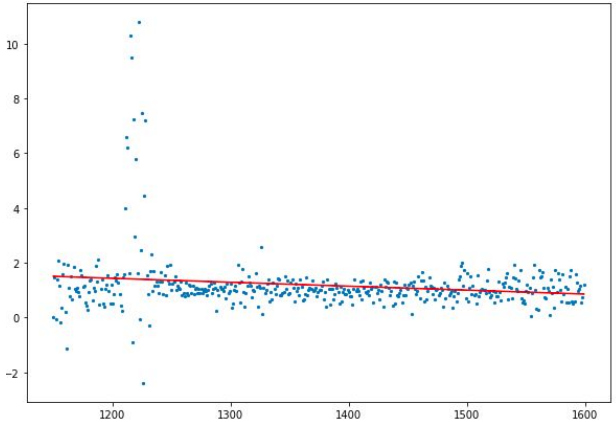
- Število čebel med
  10000 in 60000
- Dnevna temperatura zraka spomladi/poleti
  med 10°C in 35°C
- Relativna zračna vlaga med
  0 in 1 (0-100%)

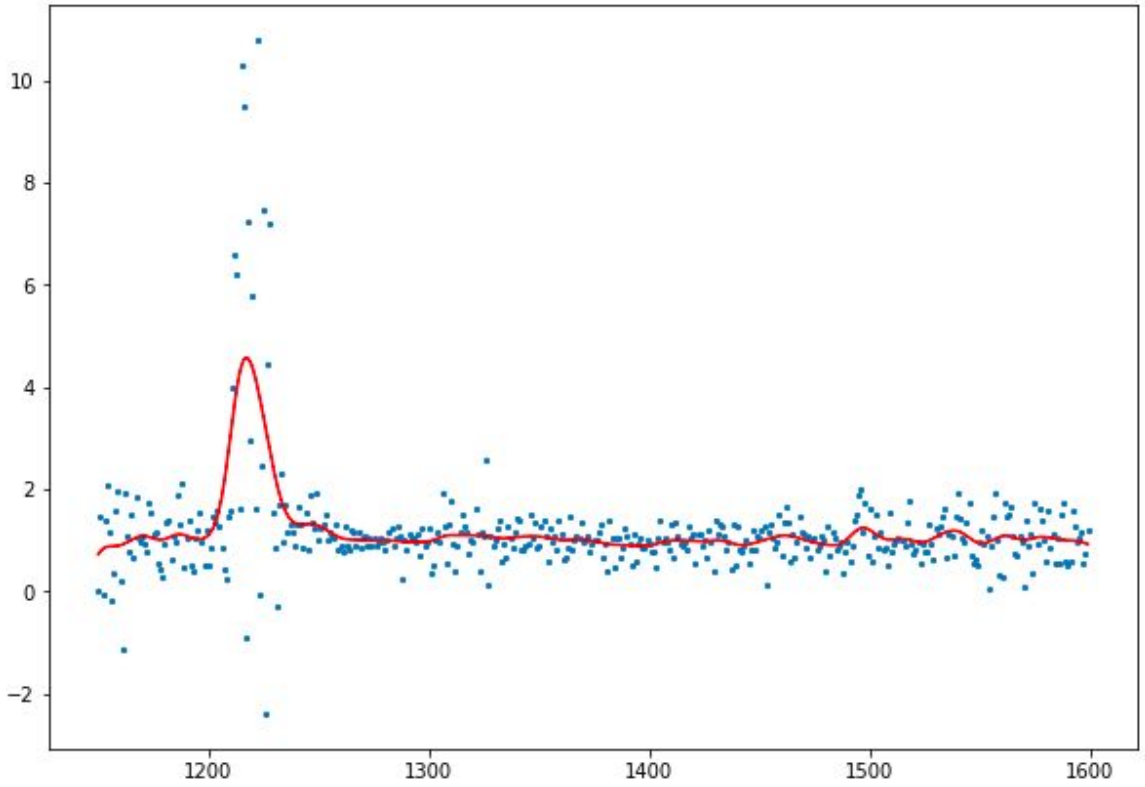
Obesity apocalypse

abcNEWS: "By 2048, all American adults would become overweight or obese."

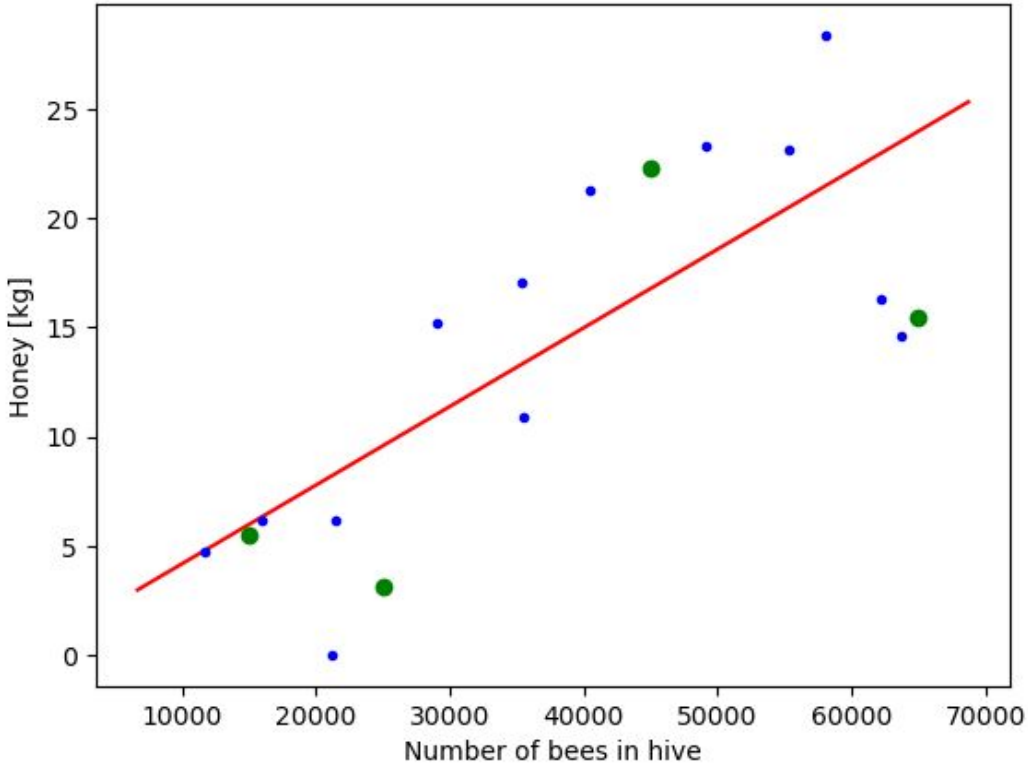


Lokalno utežena regresija (Locally weighted regression)

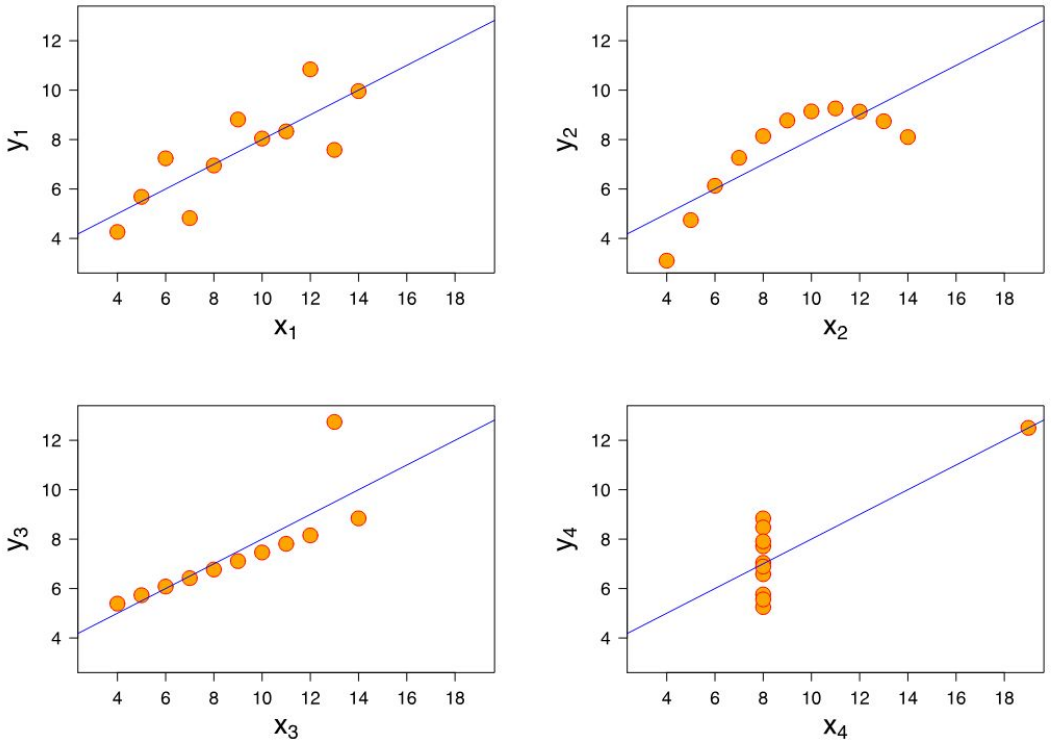




Med na panj **kNN regresija**



Nariši podatke



4 baze podatkov

skoraj **identična** statistika

zelo **različni** grafi

vir:
https://en.wikipedia.org/wiki/Anscombe%27s_quartet

Anscombe, F. J. (1973). "Graphs in Statistical Analysis". *American Statistician.* **27** (1): 17–21.