Gépi tanulás python alapokon

dmlab

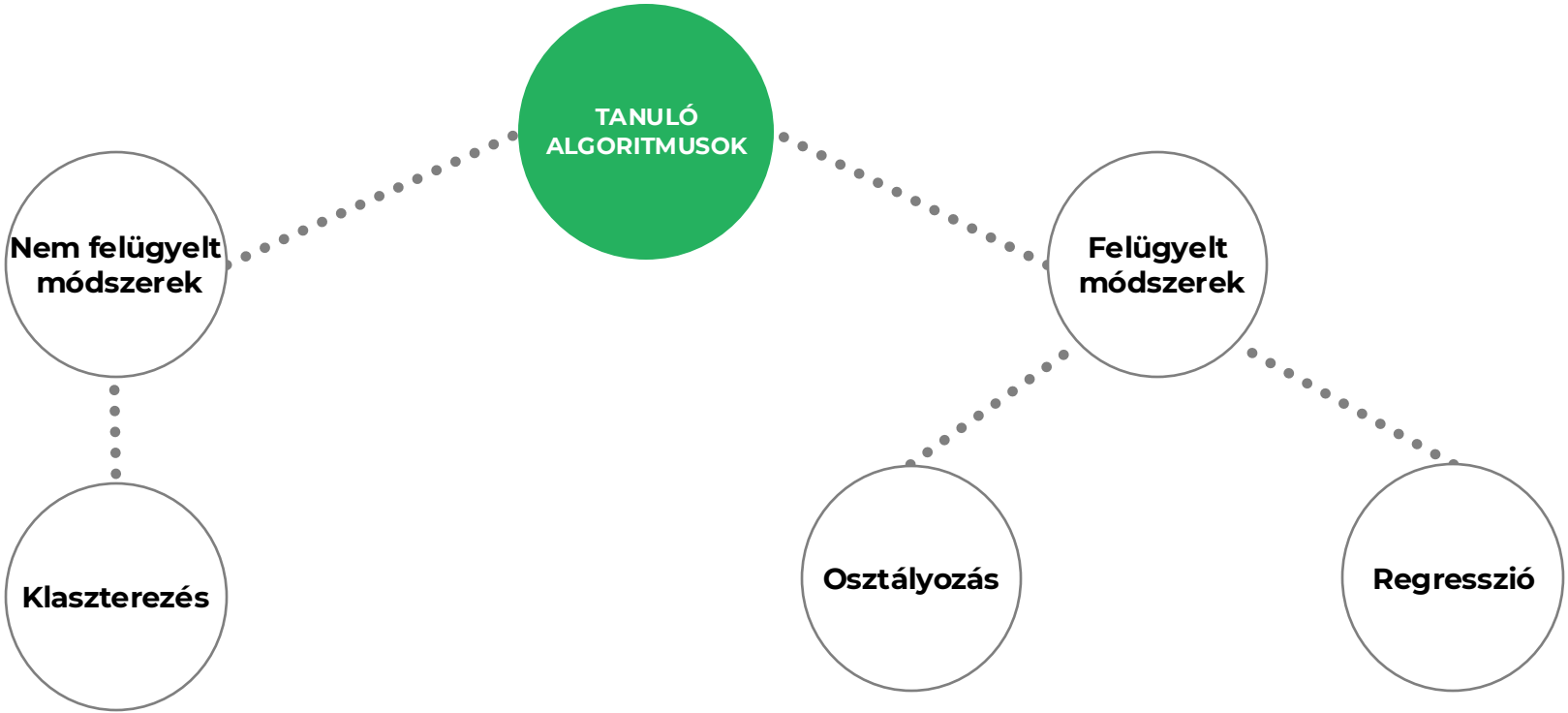
Gépi tanulási esettanulmányok

Osztályozás

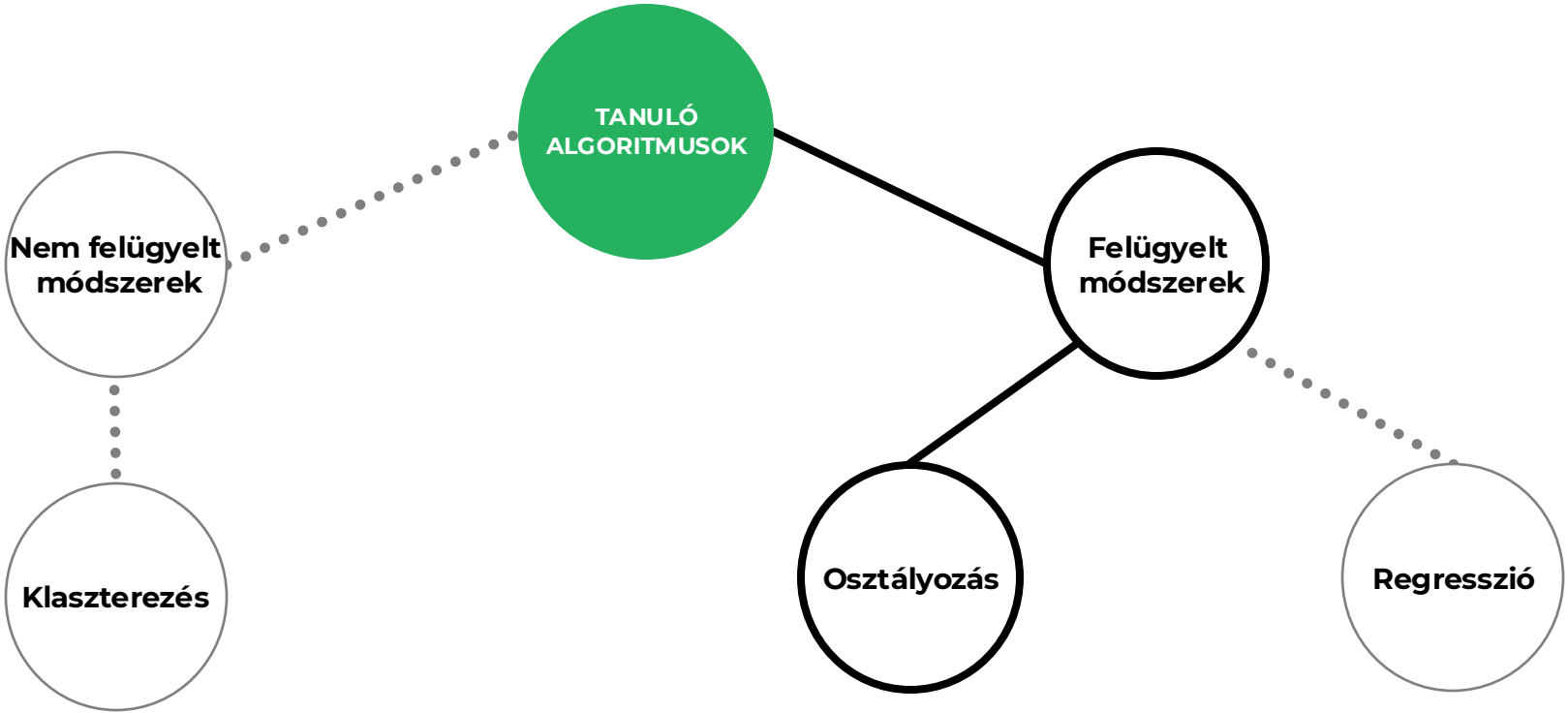
2024.09.23.

Huszti Dorottya
data scientist

MODELLEZÉSI ALAPOK



MODELLEZÉSI ALAPOK



Osztályozás

Osztályozás

Jellemzői

- Felügyelt tanulás
- Tipikus use-case: csalás detekció, szavazói döntés, churn, címkézés
- Célváltozó: diszkrét kategorikus változó
- Gépi tanulási modellek osztályozáshoz: logisztikus regresszió, döntési-fa alapú, k-nn (k darab legközelebbi szomszéd), stb..
- Kiértékelési metrikák: Accuracy, Precision, Recall, Confusion matrix

Feladat

Van egy tanító adathalmazom

... ahol ismerem a tulajdonságokat és

Tulajdonságleíró attribútumok – **bemeneti változók**

Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma
42	280 000	82	3	1
88	90 000	44	0	0
22	180 000	32	0	1
38	400 000	102	1	2

Feladat

Van egy tanító adathalmazom
... ahol ismerem a tulajdonságokat és

X				
X1	**X2**	**X3**	**X4**	**X5**

Tulajdonságleíró attribútumok – **bemeneti változók**

Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma
42	280 000	82	3	1
88	90 000	44	0	0
22	180 000	32	0	1
38	400 000	102	1	2

Feladat

Van egy tanító adathalmazom
... ahol ismerem a tulajdonságokat és
... ahol ismerem a célváltozót

X					y
X1	X2	X3	X4	X5	

Célváltozó
Target
Címke

Tulajdonságleíró attribútumok – **bemeneti változók**

Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma	Biztosítás?
42	280 000	82	3	1	OK
88	90 000	44	0	0	NO
22	180 000	32	0	1	OK
38	400 000	102	1	2	NO

Feladat

És van egy új sor, ahol
... ismerem a tulajdonságokat

Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma	Biztosítás
55	180 000	62	1	3	????

X					y
X1	**X2**	**X3**	**X4**	**X5**	

Célváltozó
Target
Címke

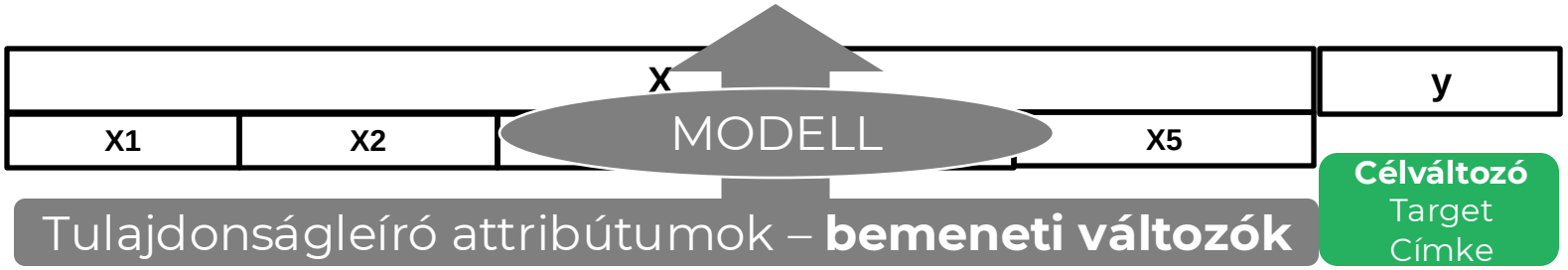
Tulajdonságleíró attribútumok – **bemeneti változók**

Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma	Biztosítás?
42	280 000	82	3	1	OK
88	90 000	44	0	0	NO
22	180 000	32	0	1	OK
38	400 000	102	1	2	NO

Feladat

És van egy új sor, ahol
... ismerem a tulajdonságokat
... modell mondja meg mekkora a célváltozó

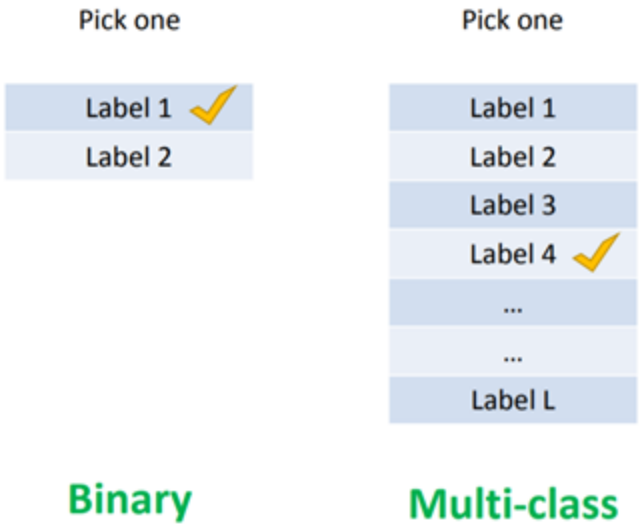
Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma	Biztosítás
55	180 000	62	1	3	????



Életkor	Jövedelem	Lakásméret	Gyerekek	Autók száma	Biztosítás?
42	280 000	82	3	1	OK
88	90 000	44	0	0	NO
22	180 000	32	0	1	OK
38	400 000	102	1	2	NO

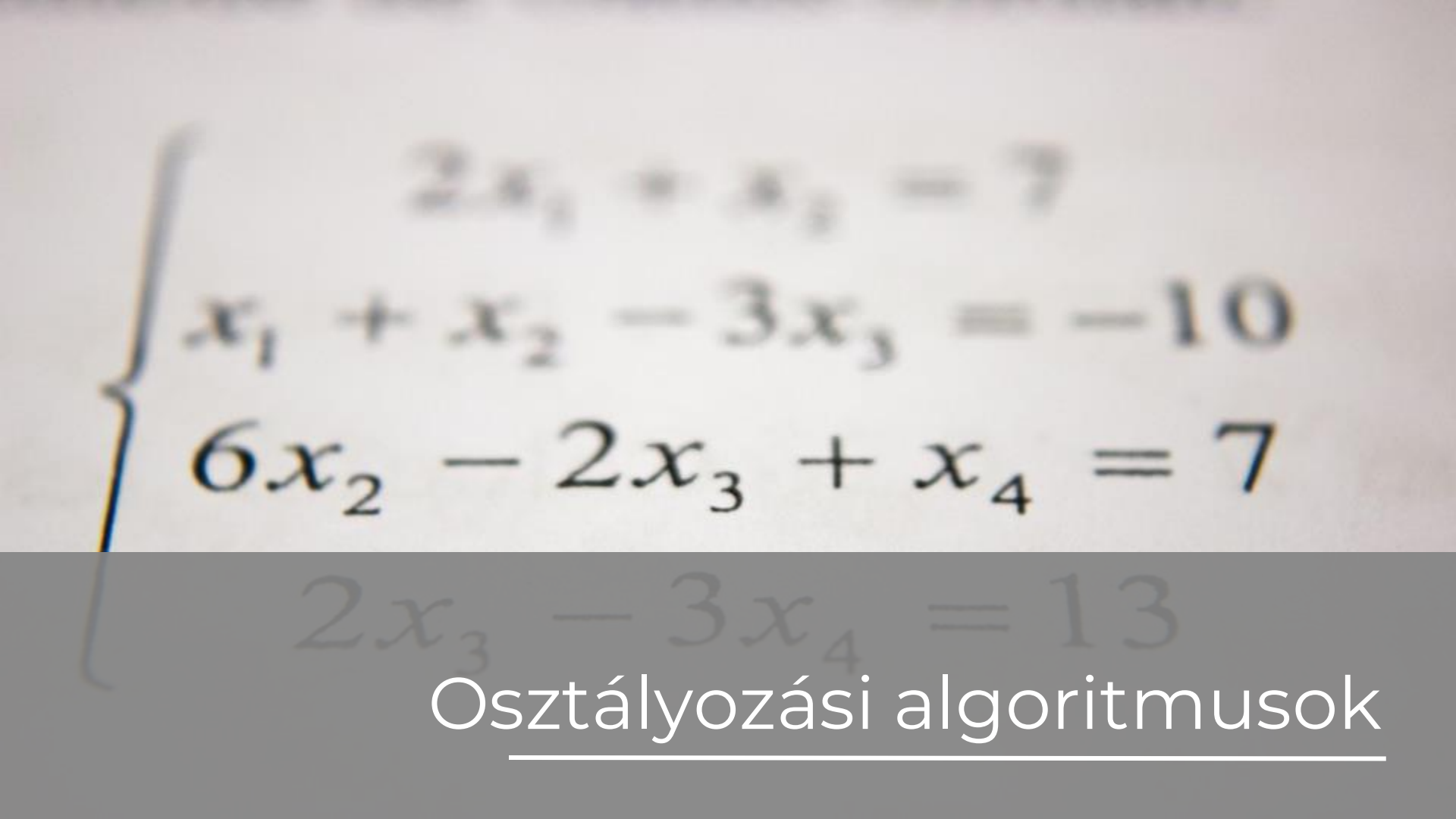
Osztályozási feladatok

- **Binary - Multi-class**
  - csalás - nem csalás
  - Kutya – macska - hal



Osztályozási feladatok

- **Binary - Multi-class**
  - csalás - nem csalás
  - Kutya – macska - hal
- **Single-label - Multi label**
  - Csalás – nem csalás
  - könyvek címkézése



Osztályozási algoritmusok

Osztályozási algoritmusok

- **Logisztikus regresszió**
- **Döntési fa**
- **K-nn**
- Összetett modellek (Ensemble models)- *(Képzés későbbi alkalmán)*:
  - Gradient Boosting Classifier
  - Random forest

LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

$$\propto = b_o + b_1 * x_1 + \cdots + b_n * x_n$$

LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

$$\propto = b_o + b_1 * x_1 + \cdots + b_n * x_n$$

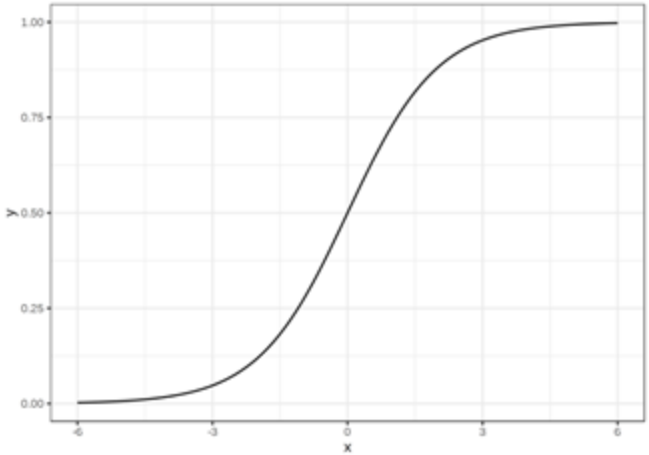
x_1	x_2	...	x_n	
area	*rooms*	*district*	*elevator*	*price*
120	4	3	1	90
50	2	3	0	45
82	3	8	1	59
46	1	13	0	40
70	3	5	0	71
65	2	3	1	54

X input features *y target*

LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

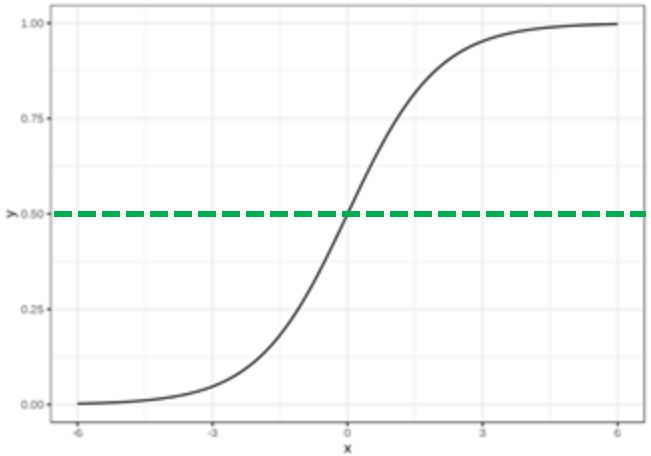
$$\propto = b_o + b_1 * x_1 + \cdots + b_n * x_n$$



LOGISZTIKUS REGRESSZIÓ

$$y = logistic(\propto) = \frac{1}{1 + \exp(-\propto)}$$

$$\propto = b_o + b_1 * x_1 + \cdots + b_n * x_n$$



LOGISZTIKUS REGRESSZIÓ

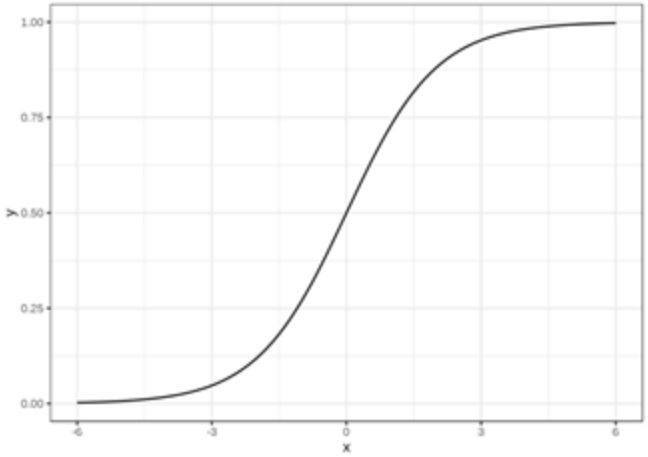
Előnyei/ Hátrányai

+ Könnyen alkalmazható, kicsi számításigény

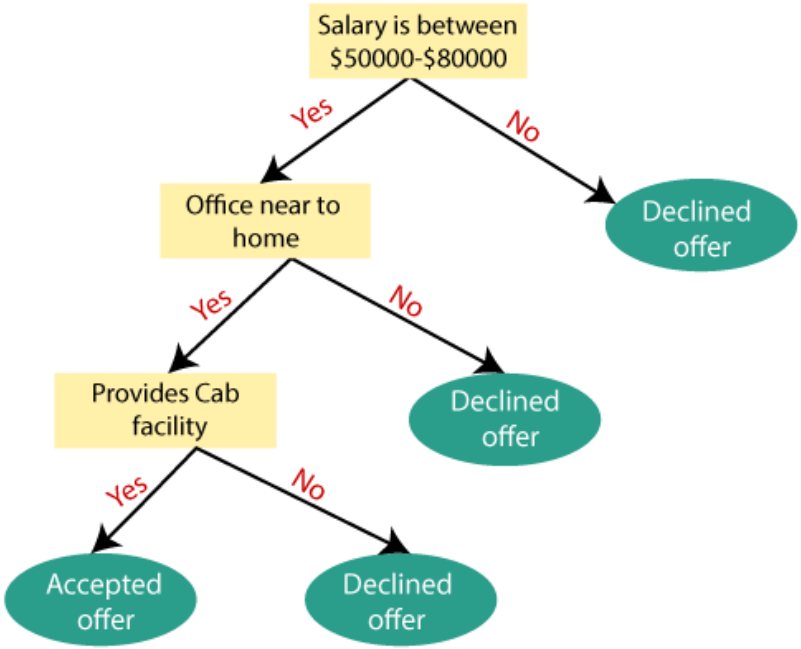
+ Jó alapmodellként szolgál

- Jó teljesítményhez szükség van a független változók kigyűjtésére

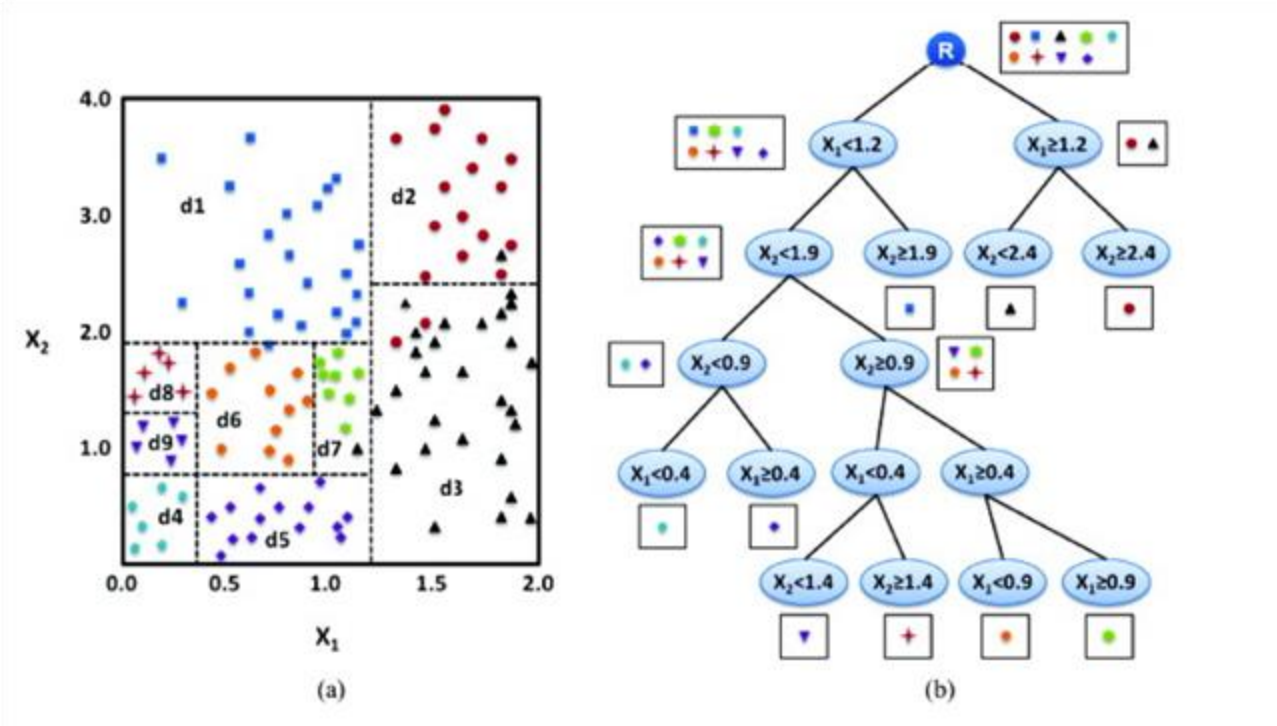
- Kevésbé könnyen interpretálható mint a döntési fa



Döntési fa


Salary is between $50000-$80000
Yes
No
Office near to home
Declined offer
Yes
No
Provides Cab facility
Declined offer
Yes
No
Accepted offer
Declined offer

DÖNTÉSI FA


4.0
3.0
2.0
1.0
0.0
0.5
1.0
1.5
2.0
X2
X1
d1
d2
d3
d4
d5
d6
d7
d8
d9
(a)
R
X1<1.2
X1≥1.2
X2<1.9
X2≥1.9
X2<2.4
X2≥2.4
X2<0.9
X2≥0.9
X1<0.4
X1≥0.4
X1<0.4
X1≥0.4
X2<1.4
X2≥1.4
X1<0.9
X1≥0.9
(b)

Döntési fa

Előnyei/ Hátrányai

+Könnyen értelmezhető

+Változó kiválasztás automatikusan megtörténik

+Relatív kis adatelőkészítést igényel

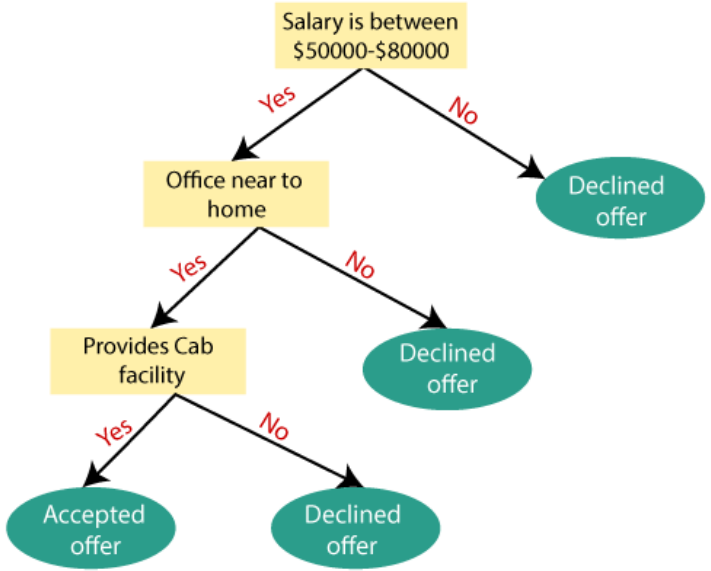
+Kevesebb adattisztítást igényel

+A paraméterek közötti nem lineáris kapcsolat nem befolyásolja a teljesítményt

-Túl tanulás

-A mohó algoritmus nem biztos hogy a globálisan legjobb döntési fát készíti el

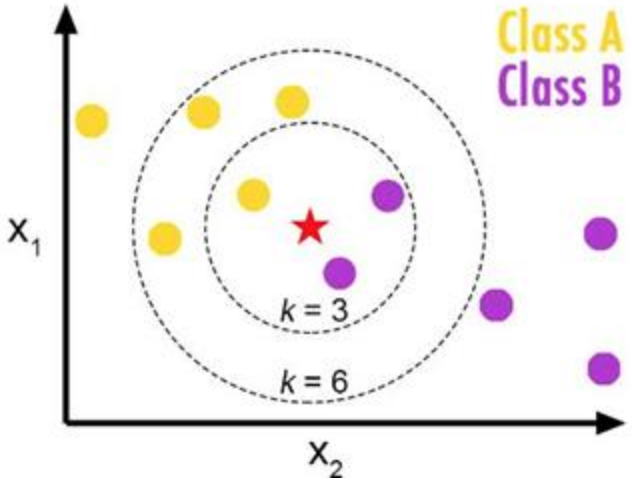
-Kiegyensúlyozatlanok lehetnek, ha egyik osztály jelentősen dominánsabb



K-NN

A lusta tanuló

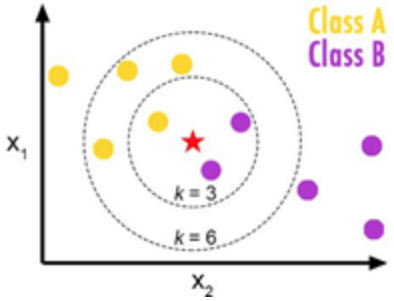
A k legközelebbi szomszédok (KNN) algoritmus egy egyszerű, könnyen megvalósítható felügyelt gépi tanulási algoritmus. A KNN algoritmus feltételezi, hogy hasonló dolgok egymáshoz közvetlen közel léteznek. Más szavakkal, hasonló dolgok közel vannak egymáshoz.



K-NN

Normalizáljuk a paramétereket:

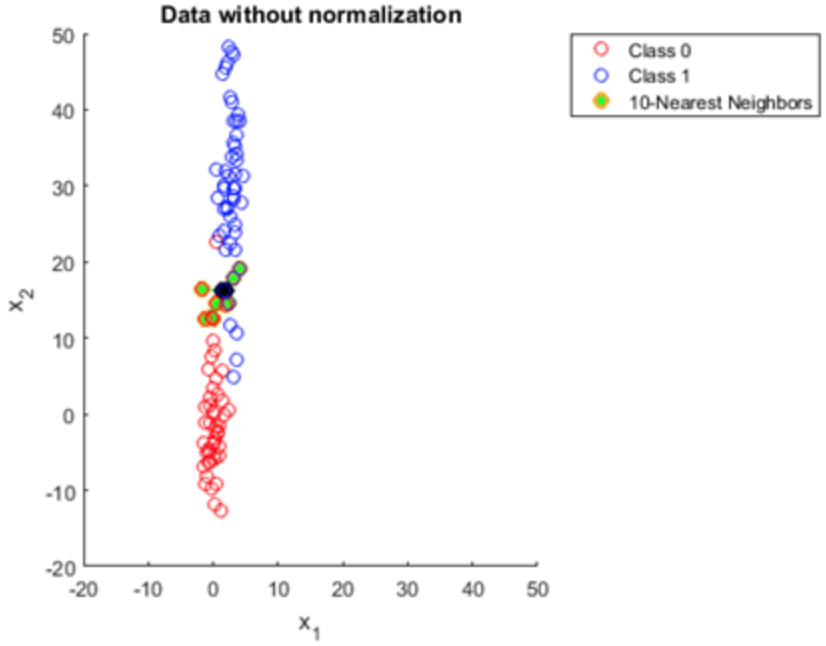
$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

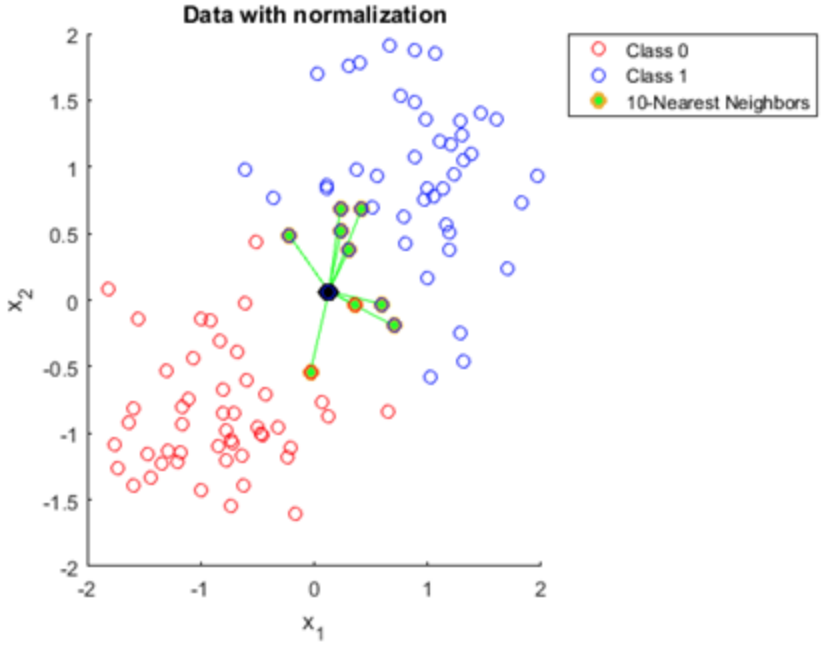


Távolság számításhoz: Eucleadian távolság

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

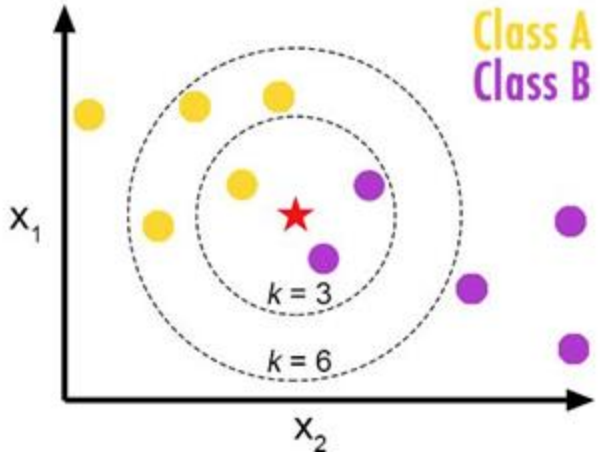
K-NN


Data without normalization
Class 0
Class 1
10-Nearest Neighbors
50
40
30
20
10
0
-10
-20
-20
-10
0
10
20
30
40
50
x2
x1


Data with normalization
Class 0
Class 1
10-Nearest Neighbors
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2
x2
x1

Hogyan válasszuk meg k-t?

- Amint a K értékét 1-re csökkentjük, predikcióink kevésbé stabilak lesznek. Például, ha K == 1 és 99 pont a 100-ból sárga, de a legközelebbi lila, akkor lilát predikálunk
- Fordítva, ahogy növeljük a K értékét, predikcióink stabilabbá válnak a többségi szavazás / átlagolás miatt, és így nagyobb valószínűséggel fognak pontosabb előrejelzéseket tenni.
- Azokban az esetekben, amikor a címkék között többségi szavazást folytatunk, K-t általában páratlan számként definiáljuk, hogy elkerüljük a tiebreakert.



K-NN

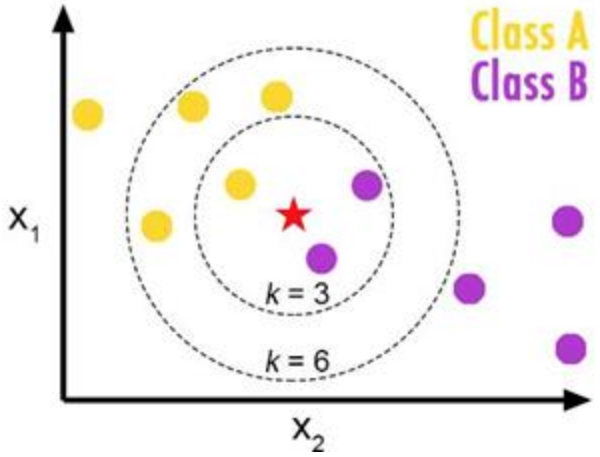
Előnyei/ Hátrányai

+ Az algoritmus egyszerű és könnyen megvalósítható.

+ Nincs szükség modell felépítésére, több paraméter hangolására vagy további feltételezésekre.

+ Az algoritmus sokoldalú. Használható osztályozáshoz, regresszióhoz és kereséshez.

- Az algoritmus jelentősen lassabb lesz, ahogy a példák és / vagy előrejelzők / független változók száma növekszik.



KIÉRTÉKELÉSI METRIKÁK

Kiértékelési metrikák

Metrikák:

- Konfúziós mátrix (Confusion matrix)
- Pontosság (Accuracy)
- Fedés (Recall)
- Pontosság?! (Precision)

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

Confusion matrix

Jellemzői

- Átfogó képet ad a modellünk teljesítményéről
- A további metrikák ebből vezethetők le

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

Accuracy

Jellemzői

- Népszerű és könnyen interpretálható metrika
- Sok esetben csalóka lehet

Accuracy = (TP+TN)/(TP+FP+FN+TN)

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

Precision

Jellemzői

Hány kiválasztott elem releváns

- A Precision egy helyes kiértékelési metrika, mikor szeretnénk biztosak lenni az előrejelzésünket illetően.
- Pl.: Ha egy olyan rendszert építünk, melynek segítségével szeretnénk a kártyatulajdonosok hitelkeretét csökkenteni, biztosra kell mennünk, mert egy nem megalapozott limit csökkentés felhasználói elégedetlenséget eredményezhet

Precision = (TP)/(TP+FP)

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

Recall

Jellemzői

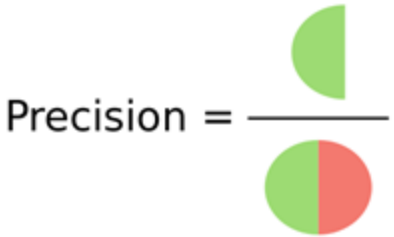
Hány releváns elem lett kiválasztva

- A Recall egy helyes metrika, mikor az a célunk hogy a lehető legtöbb pozitív elemet kiszűrjük.
- Pl.: Ha egy olyan rendszert építünk , melynek feladata a tumordetekció a legfontosabb hogy az összes esetet kiszűrjük, még akkor is ha néha néha olyanokat is kiválasztunk , melyekbe nem vagyunk biztosak

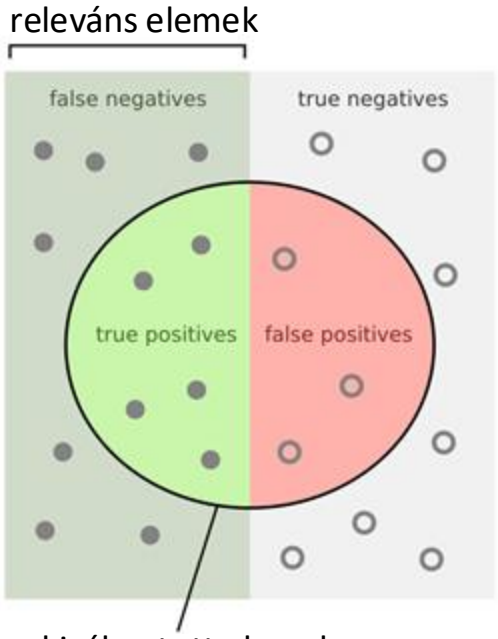
Recall = (TP)/(TP+FN)

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

Precision vs Recall



A kiválasztottak közül mennyi releváns

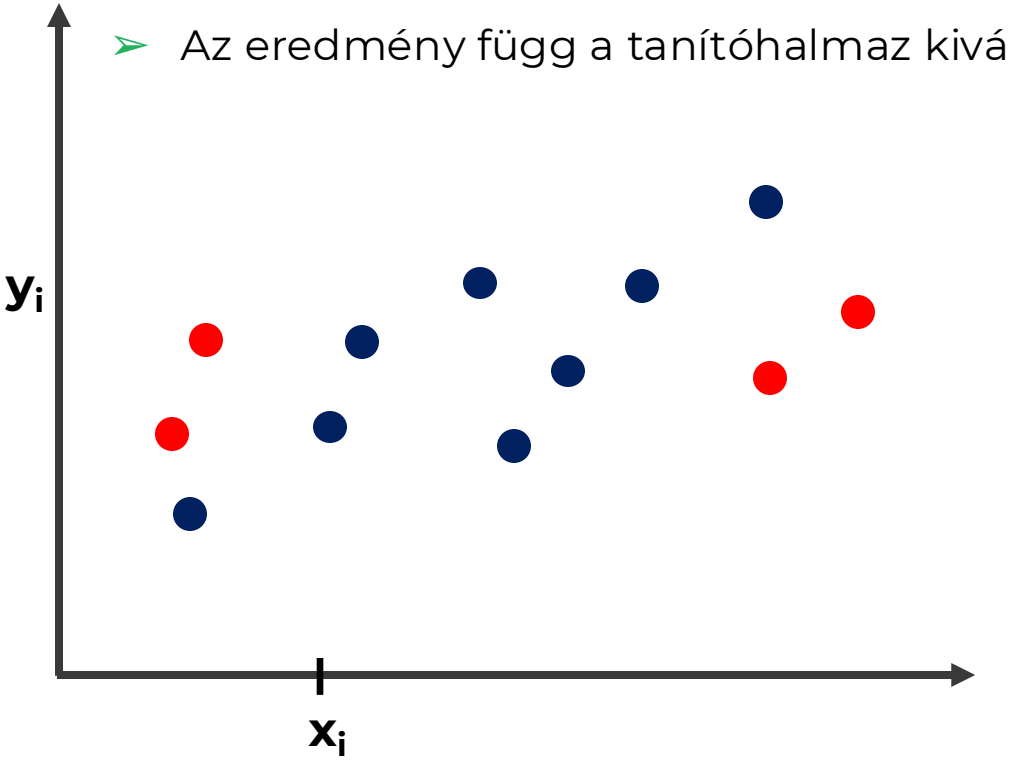


A releváns elemek hány százalékát találtuk meg

KIÉRTÉKELÉSI MÓDSZEREK

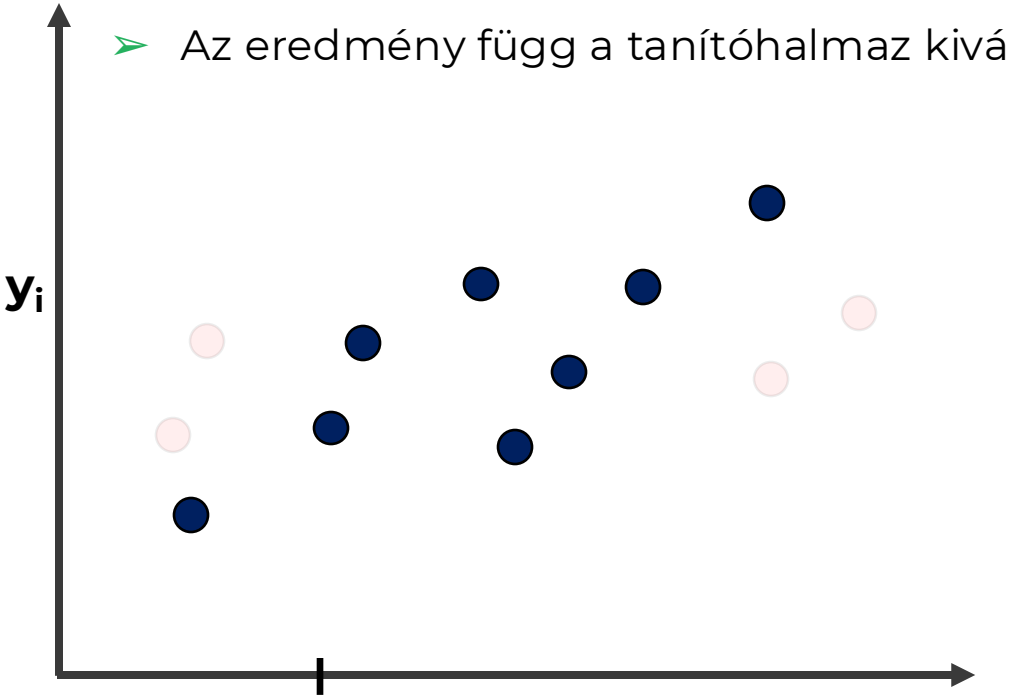
TRAIN-VALIDATION-TEST

- Az eredmény függ a tanítóhalmaz kiválasztásától

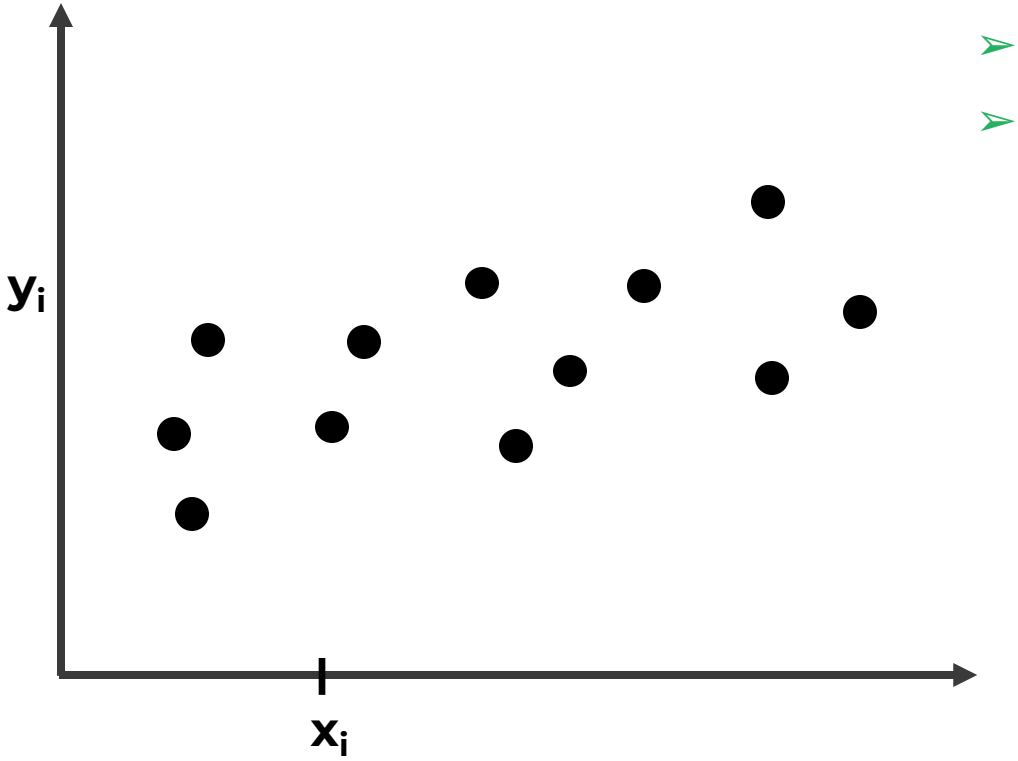


TRAIN-VALIDATION-TEST

- Az eredmény függ a tanítóhalmaz kiválasztásától

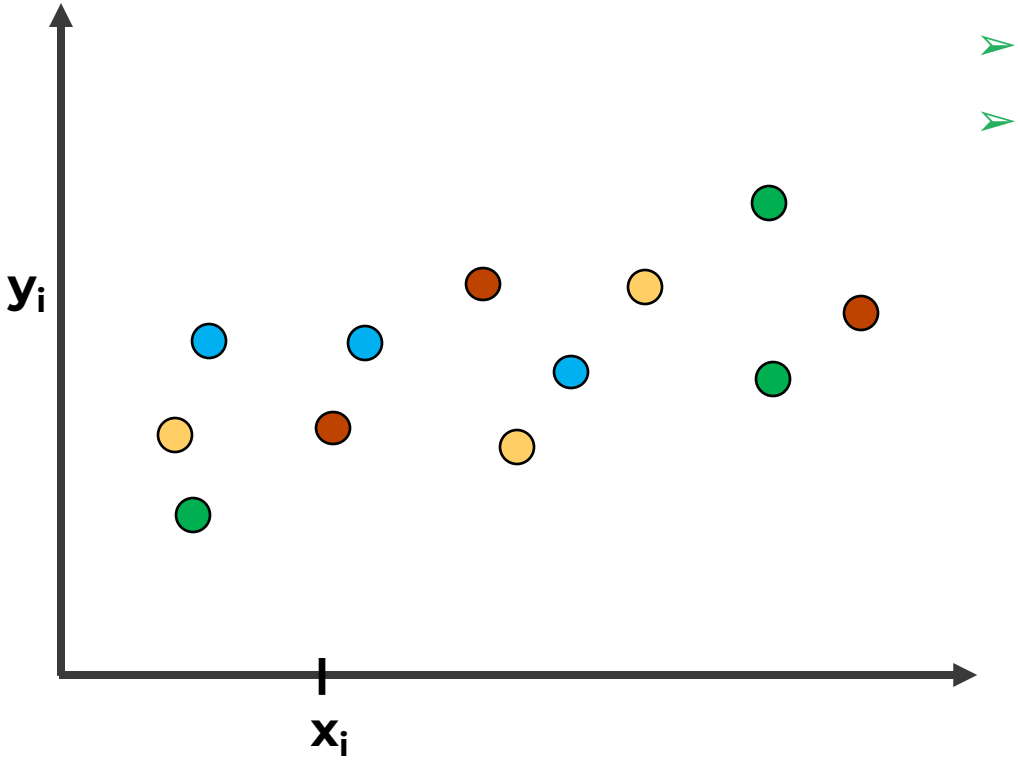


KERESZTVALIDÁCIÓ



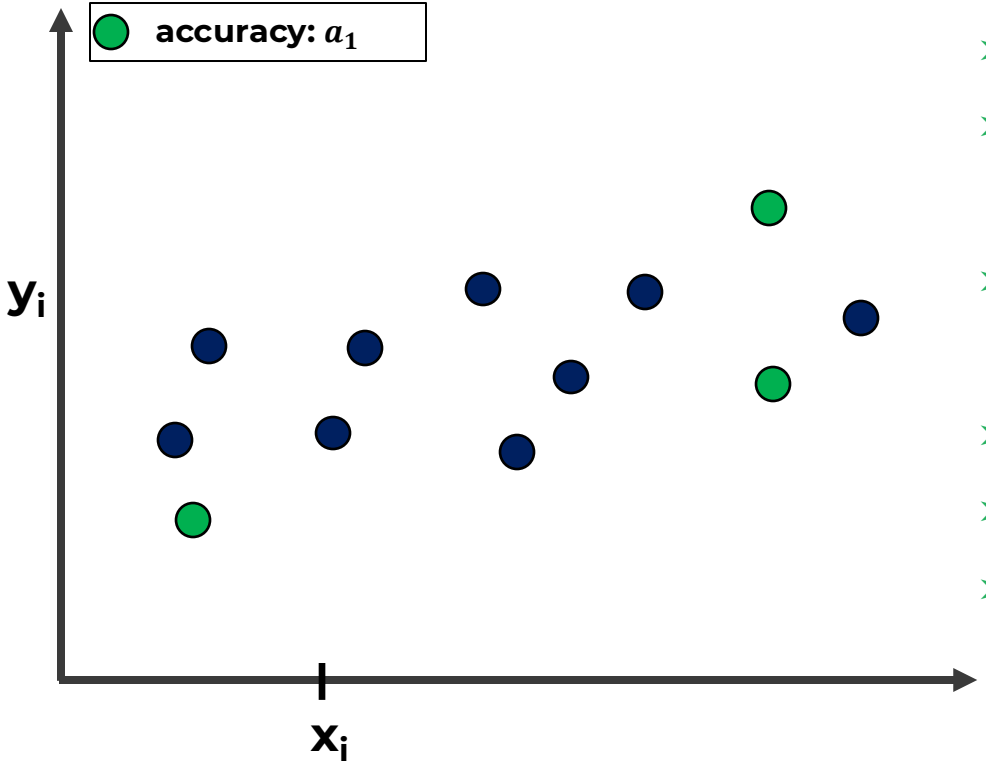
- K-fold keresztvalidáció
- Legyen k = 4

KERESZTVALIDÁCIÓ



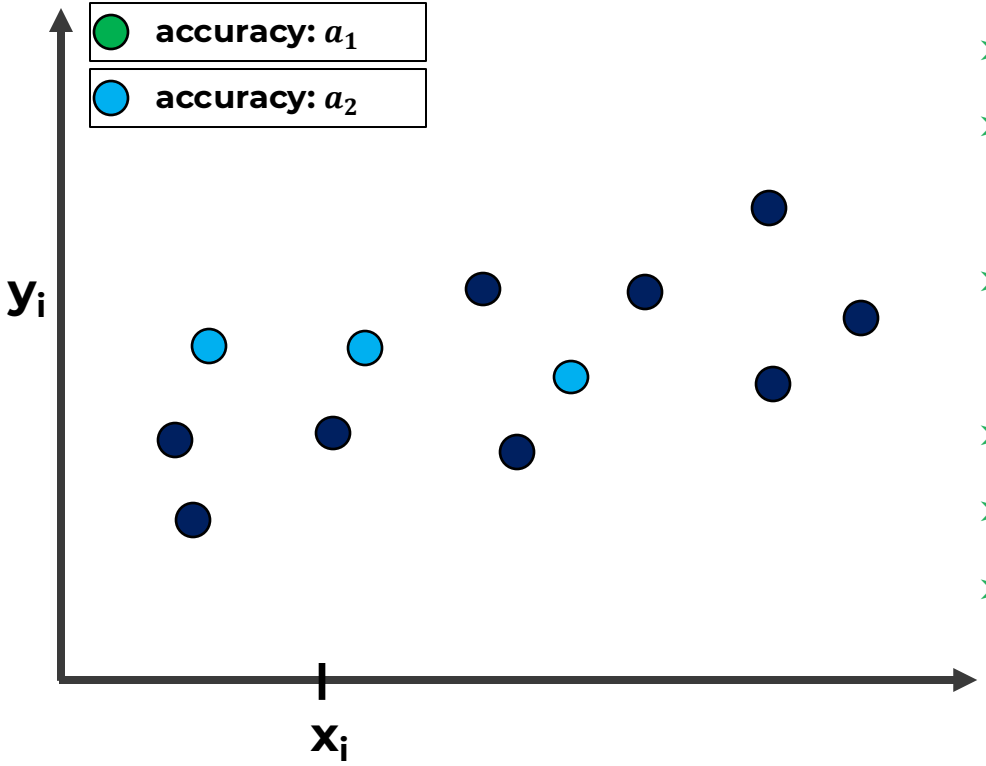
- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt

KERESZTVALIDÁCIÓ



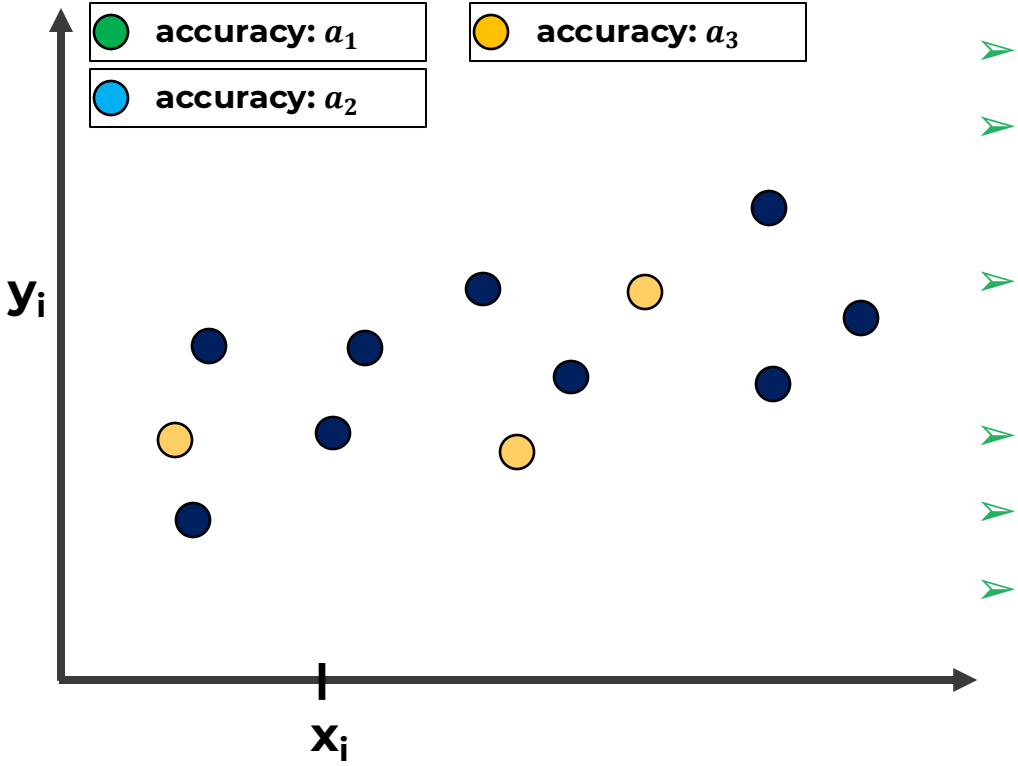
- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt
- Egyik csoportot jelöljük ki tesztnek
- Többin tanítsunk
- Értékeljük ki
- Ismételjük összes csoporttal

KERESZTVALIDÁCIÓ



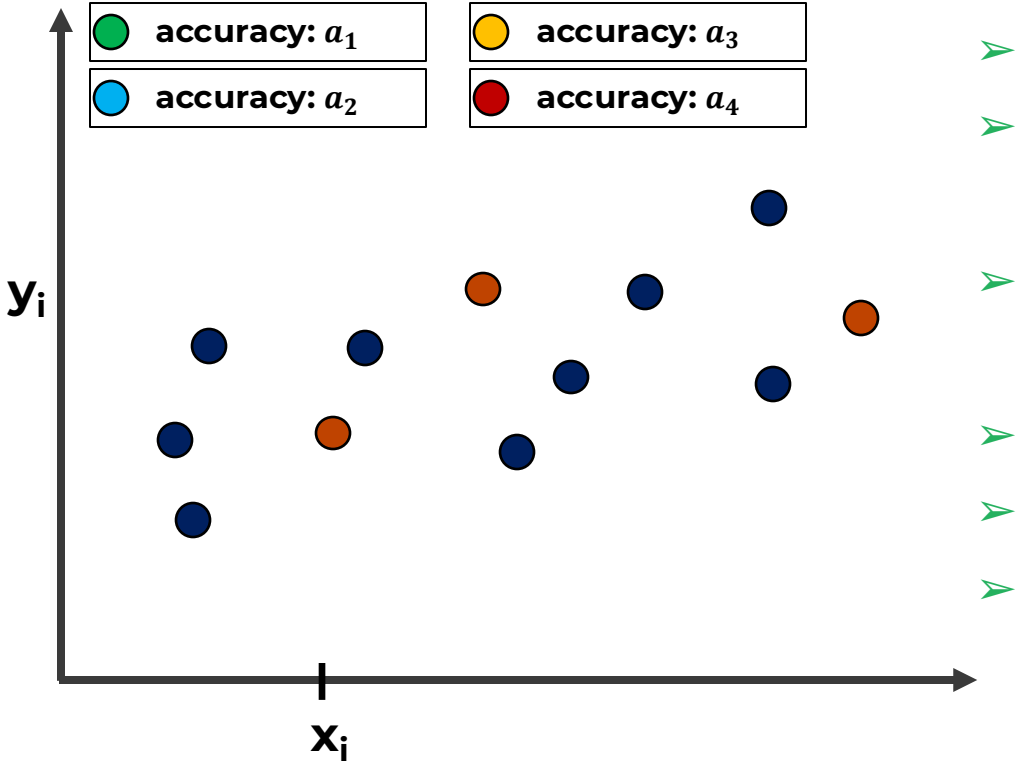
- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt
- Egyik csoportot jelöljük ki tesztnek
- Többin tanítsunk
- Értékeljük ki
- Ismételjük összes csoporttal

KERESZTVALIDÁCIÓ



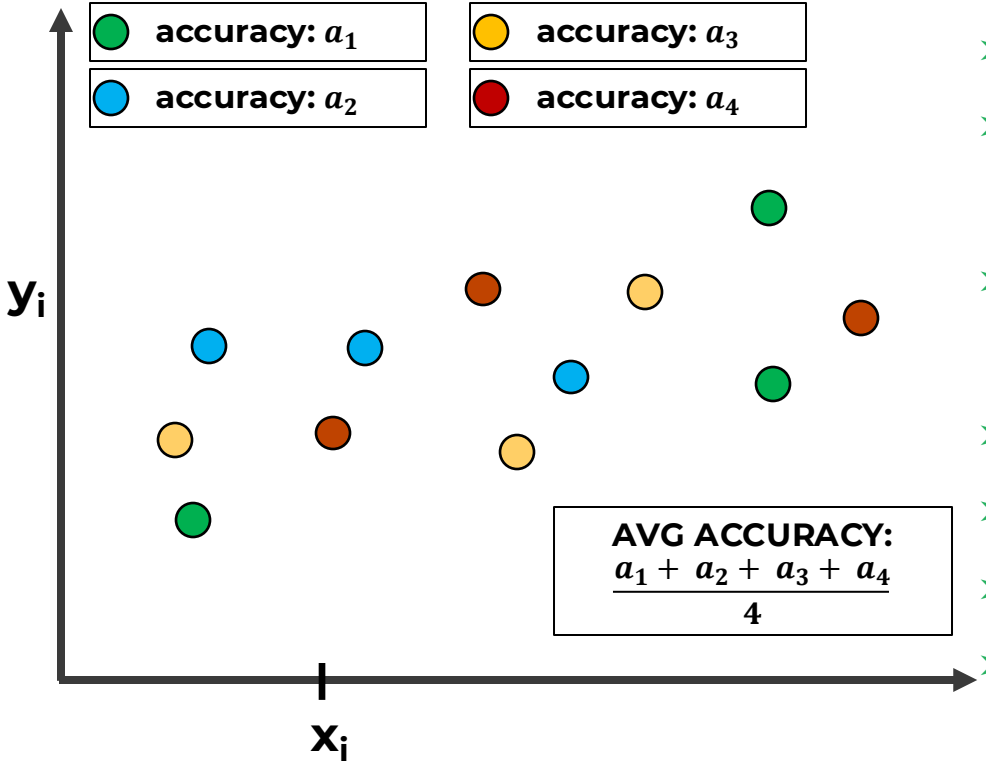
- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt
- Egyik csoportot jelöljük ki tesztnek
- Többin tanítsunk
- Értékeljük ki
- Ismételjük összes csoporttal

KERESZTVALIDÁCIÓ



- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt
- Egyik csoportot jelöljük ki tesztnek
- Többin tanítsunk
- Értékeljük ki
- Ismételjük összes csoporttal

KERESZTVALIDÁCIÓ



- K-fold keresztvalidáció
- Legyen k = 4 -> osszuk 4 véletlen csoportra az adathalmazt
- Egyik csoportot jelöljük ki tesztnek
- Többin tanítsunk
- Értékeljük ki
- Ismételjük összes csoporttal
- Átlagoljuk az eredményeket

KERESZTVALIDÁCIÓ

- Az eredmény nem függ a tanítóhalmaz kiválasztásától