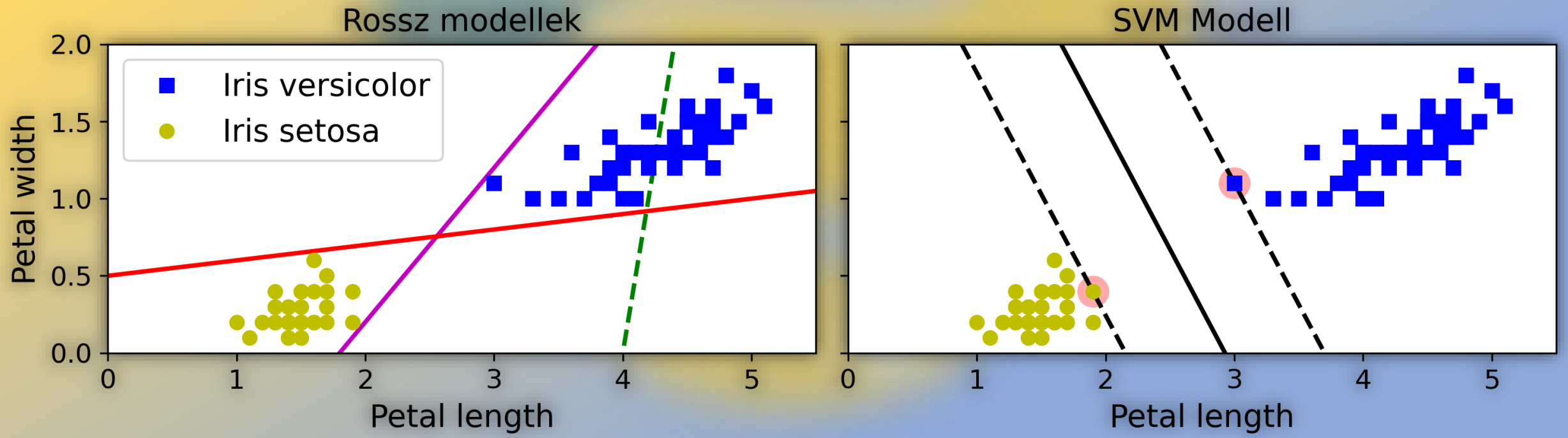
6. Előadás
Tartó vektor gépek
Lágy- és keménymargós osztályozás
Kernelizált SVM

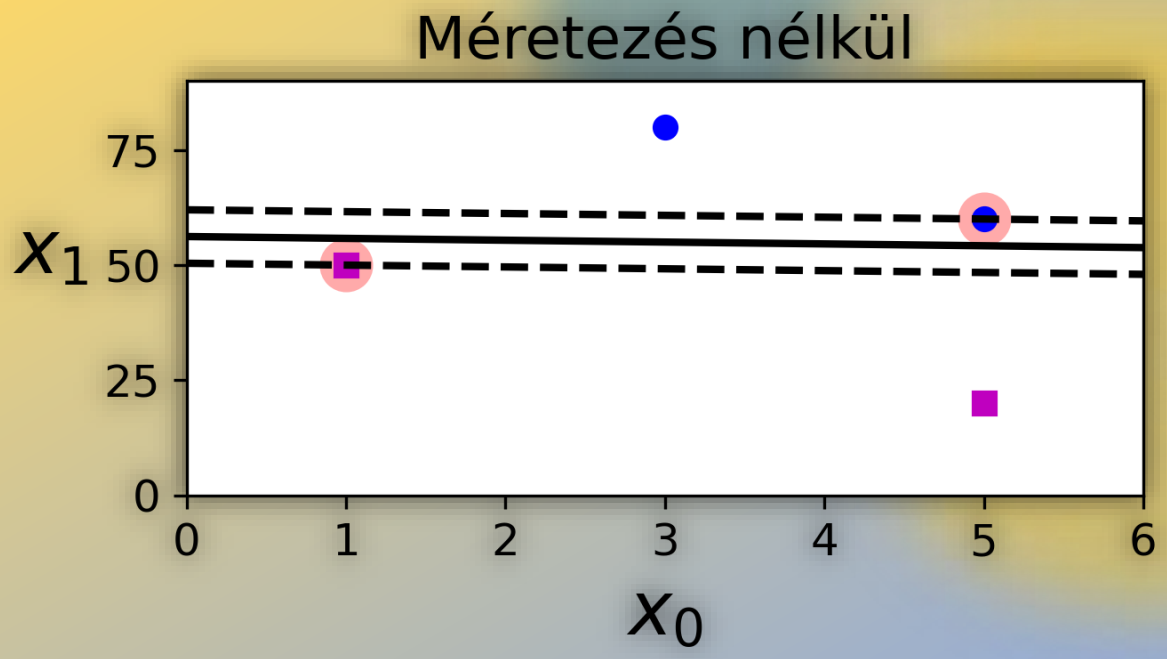
A tartó vektor gépek (SVM) alap elképzelése

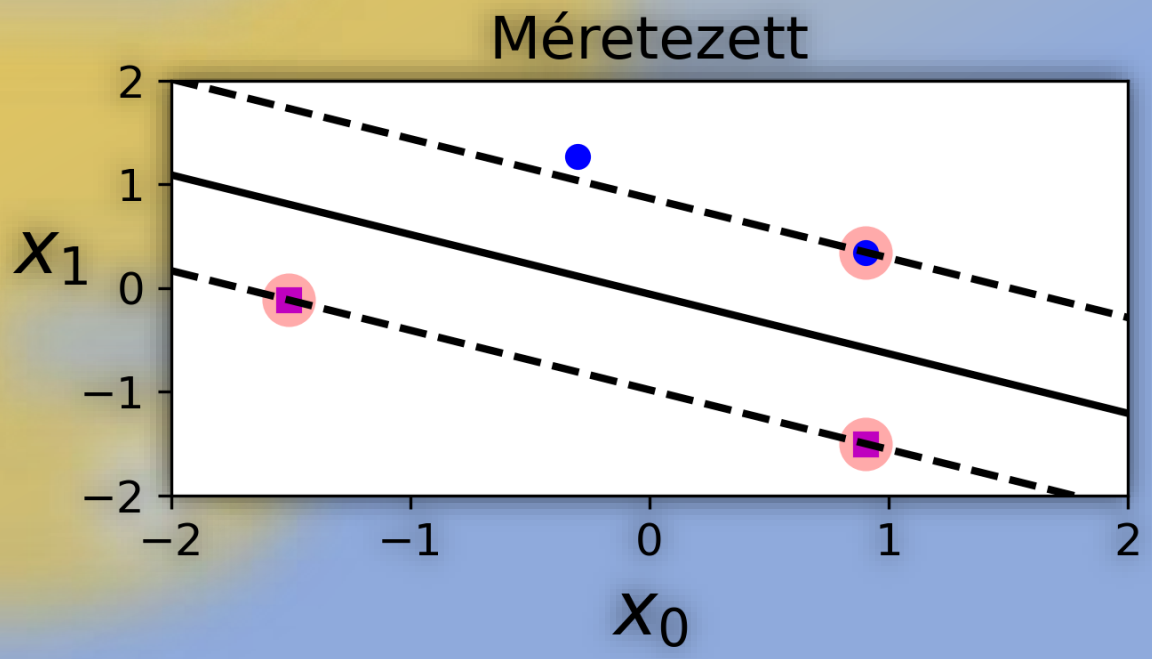
- Az SVM-ekre gondolhatunk úgy, mint a lehető legnagyobb út illesztésére osztályok között. Ez a **nagymargós osztályozás**.
- Az úton kívül hozzáadott extra mintaegyedek nem befolyásolják a döntési határt: az utat a csoportok szélén lévő mintaegyedek „tartják”.
- A margó egyik oldalán A, másik oldalán B osztály lesz a predikció:



Adatelőkészítés

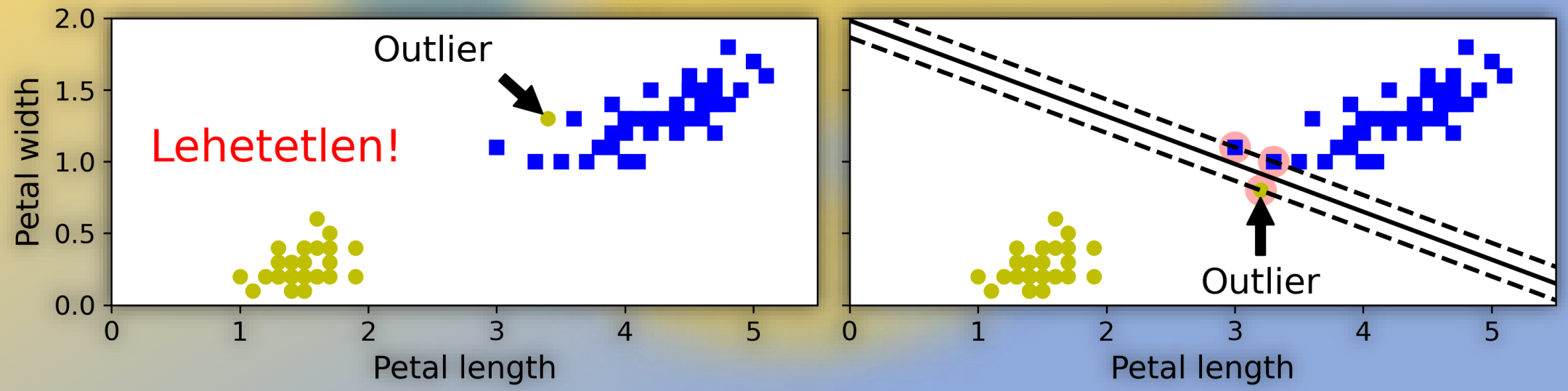
- A tartó vektor gépek rendkívül szenzitívek az adatok méretezésére.
- Méretezés után minden változó ugyanakkora mértékben járul hozzá a predikcióhoz.
- Méretezés előtt az x_1 tengely jóval nagyobb, mint az x_0, ezért a lehető legnagyobb út szinte vízszintes.





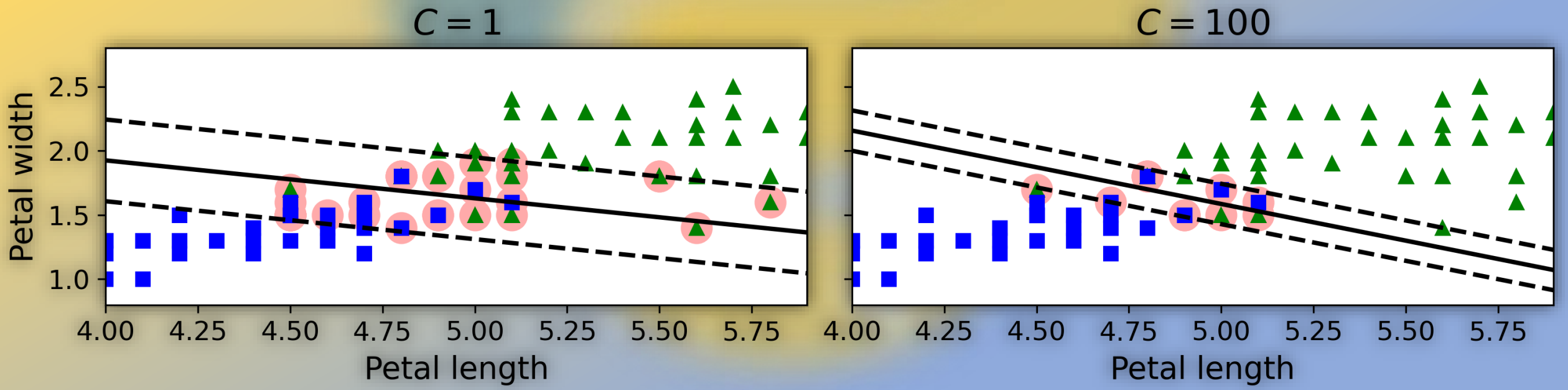
Keménymargós osztályozás

- Amennyiben megtesszük kitételnek, hogy minden mintaegyed kívül essen az úton a jobb oldalra, beszélünk **keménymargós osztályozásról**.
- Ezzel az a probléma, hogy csak akkor működik, amikor az adathalmazok lineárisan szeparálhatóak, és emellett nagyon szenzitívek a kiugró értékekre. Amikor működik, is van esélye, hogy torz modellt eredményez.
- Ennek kiküszöbölésére olyan modell kell, ami rugalmas, mégis minimumon tartja **margósértéseket**.



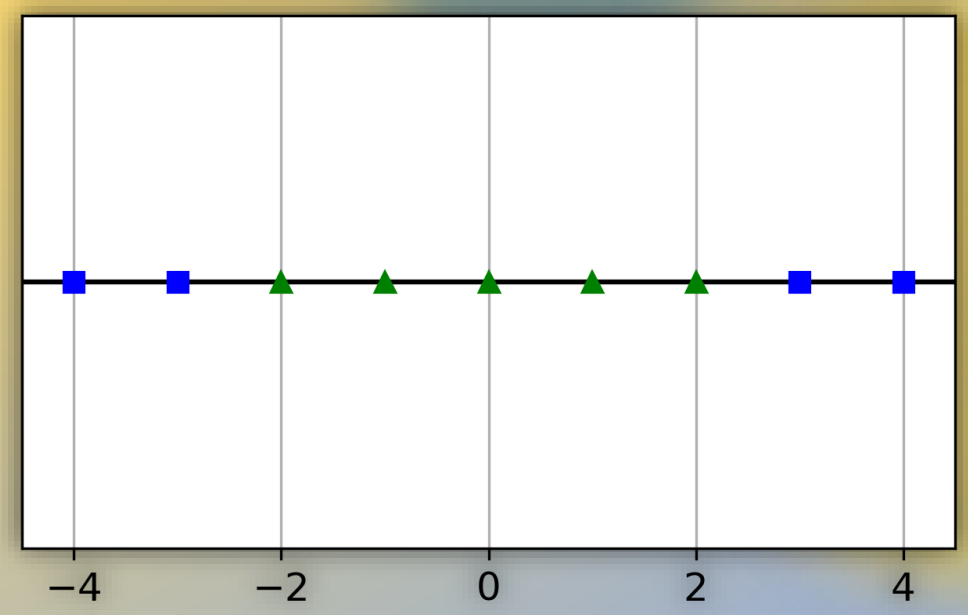
Lágymargós osztályozás

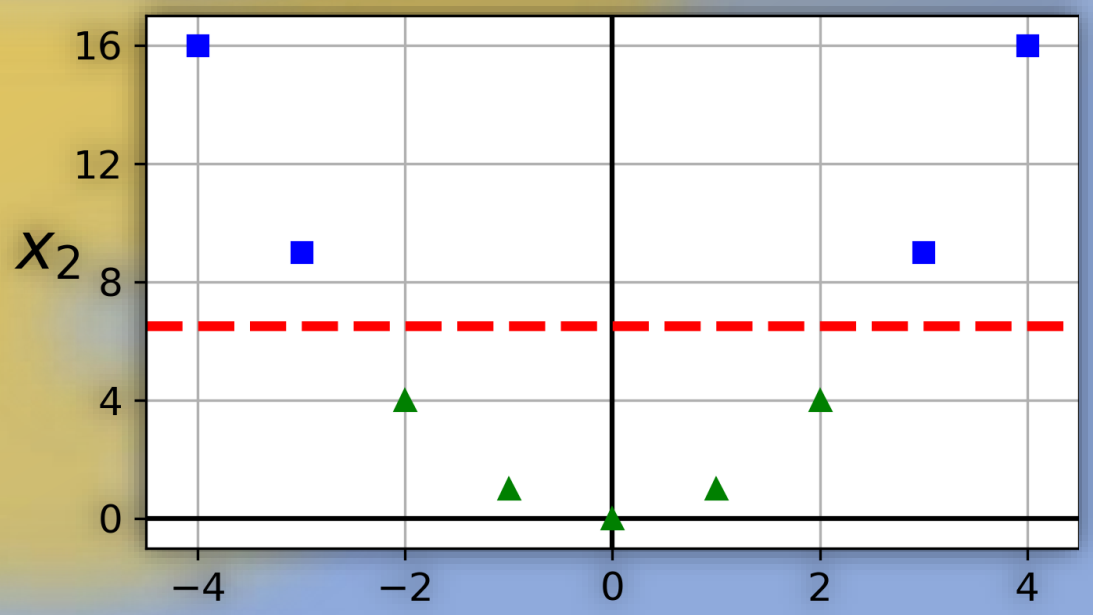
- A margósértések és a rosszul generalizáló modellek közötti optimum megtalálására hivatottak a **lágymargós** SVM modellek.
- Ennek a mértéknek a szabályozására hivatott a C hiperparaméter. A kisebb C érték szélesebb úthoz, de több margósértéshez vezet, a nagyobb C kevesebb margósértést, de rosszabbul generalizáló modellhez vezet.



Nemlineáris SVM osztályozás

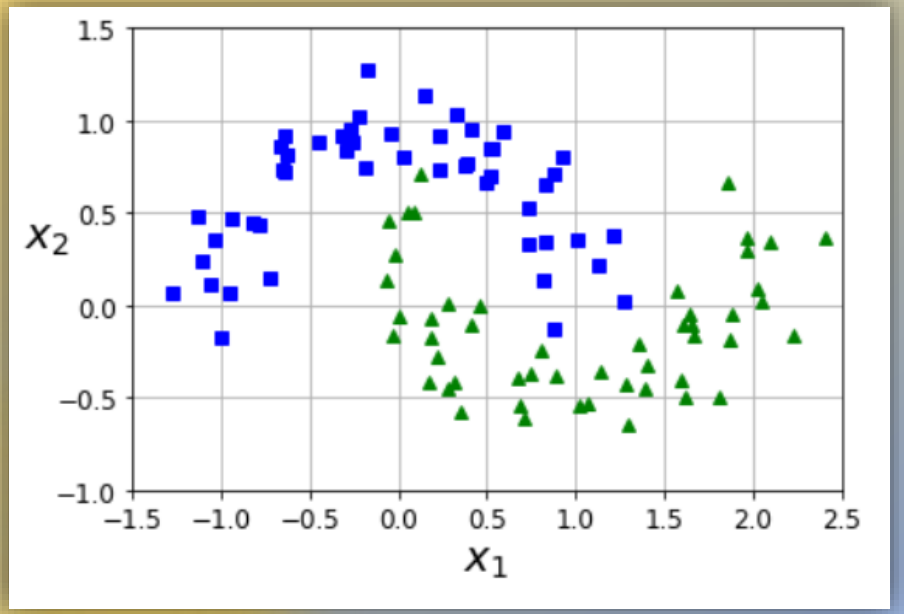
- Habár az SVM-ek meglepően jól működnek lineárisan szeparálható adathalmazok esetén, a valóságban az ilyen adathalmazok főnyereménynek számítanak.
- Egy hozzáállás a nemlineáris szeparáláshoz az, ha a meglévő jellemzőknek felvesszük a polinomikus transzformációit. Ekkor az adathalmaz (nem minden esetben) lineárisan szeparálhatóvá válik.





NL-SVM a gyakorlatban

- A szükséges transzformációkhoz a csővezeték:
  - PolynomialFeatures: polinomikus jellemzők felvétele
  - StandardScaler: jellemzők méretezése
  - LinearSVC: SVM osztályozó

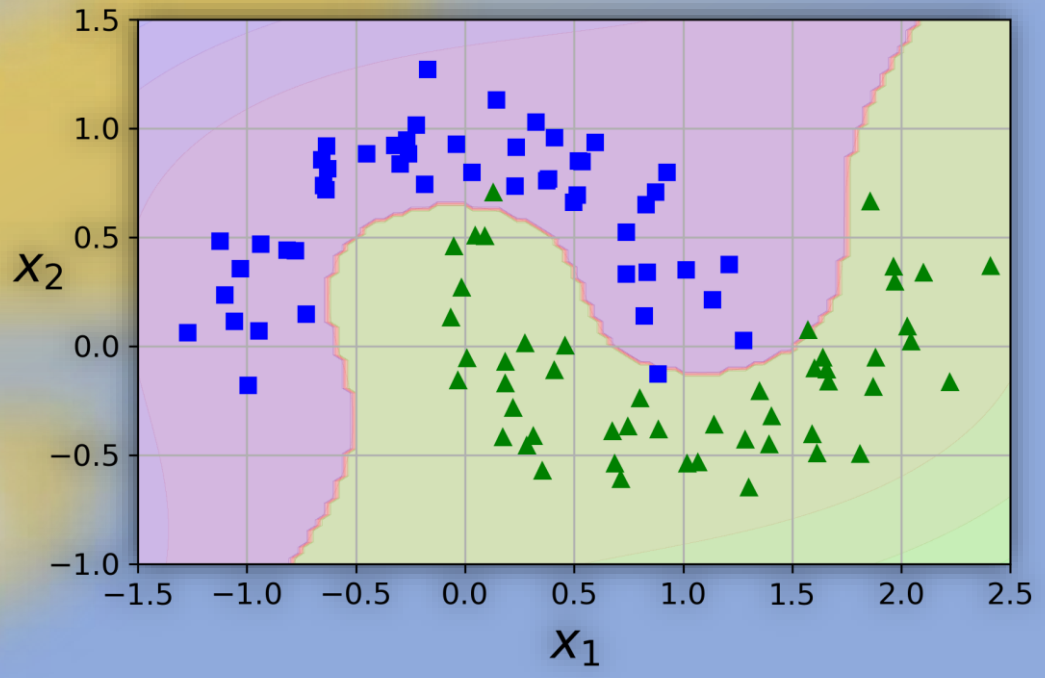



```
from sklearn.datasets import make_moons
X, y = make_moons(n_samples=100, noise=0.15, random_state=42)
```

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures

polynomial_svm_clf = Pipeline([
        ("poly_features", PolynomialFeatures(degree=3)),
        ("scaler", StandardScaler()),
        ("svm_clf", LinearSVC(C=10, loss="hinge", random_state=42))
    ])

polynomial_svm_clf.fit(X, y)
```

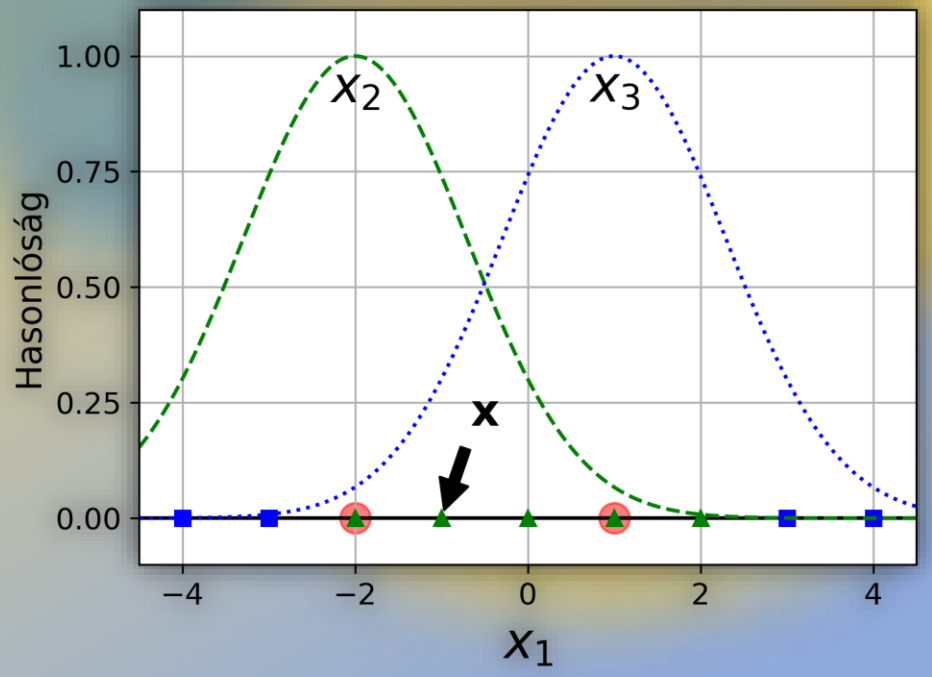


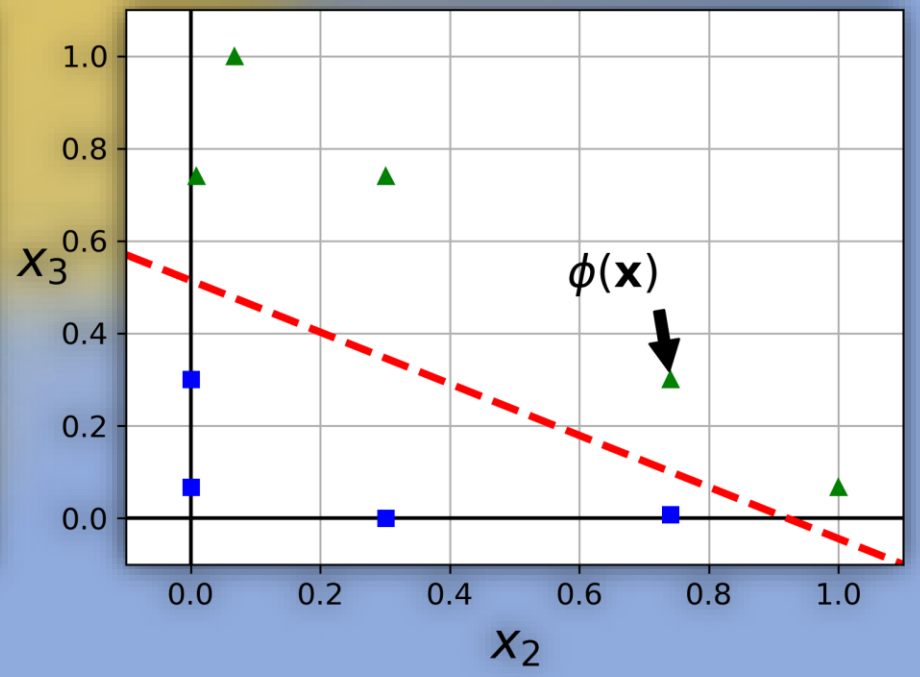
Hasonlósági függvények

- A hasonlósági függvény azt képezi le, hogy egy adott pont mennyire hasonlít egy **tájékozódási ponthoz**.
- Adjunk az előző, 1D halmazhoz két ilyen pontot $x_1 = -2$ és $x_2 = 1$ helyeken.
- Definiáljuk a hasonlósági függvényt: legyen a Gauss-i Radiális Bázis Függvény $\gamma = 0.3$ paraméterrel:

$$\phi_\gamma(\mathbf{x}, \ell) = \exp\left(-\gamma \| \mathbf{x} - \ell \|^2\right)$$

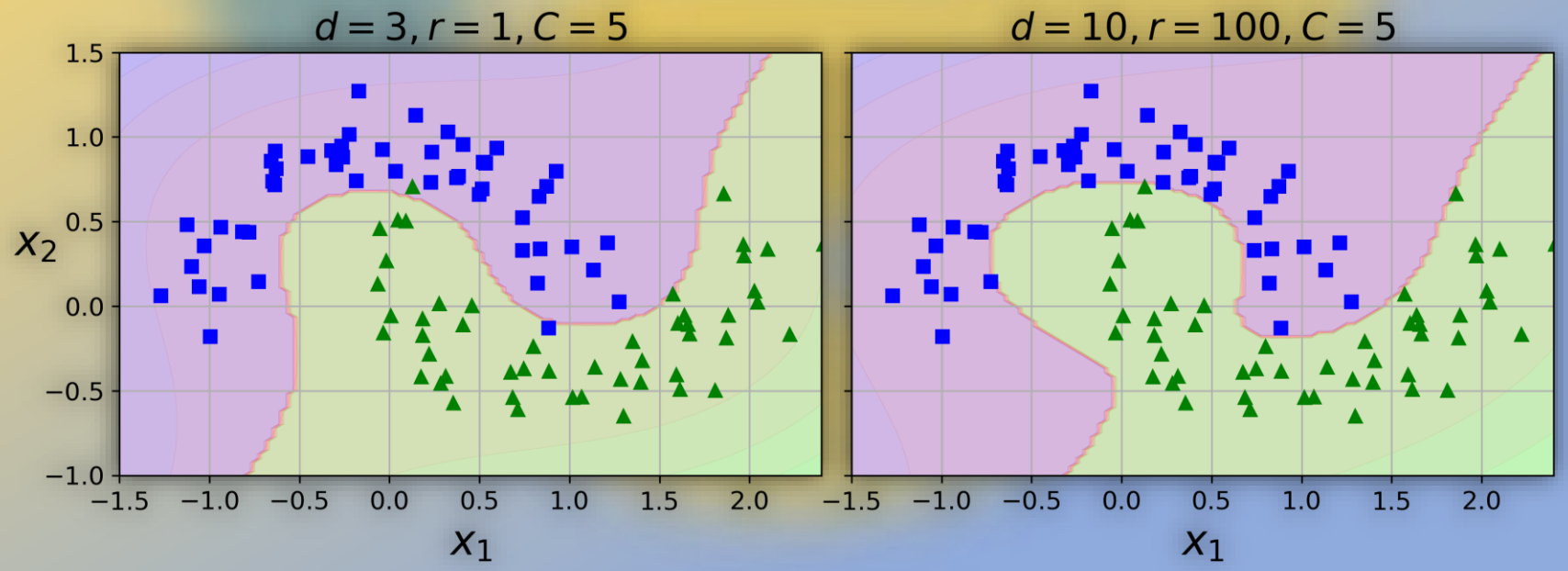
- A függvény egy haranggörbe 0 … 1 értékkészlettel aszerint, hogy a pont hasonló (1=tájékozódási pont) vagy nem hasonló hozzá (0).





A polinomikus kernel

- A polinomikus jellemzők hozzáadása alacsony polinomikus szinten nem tud komplex adathalmazokkal dolgozni, magas szinten pedig lassítja a modellt.
- Szerencsére az SVM-ek esetében lehetőség van egy matematikai technika, a **kernel trükk** bevetésére.
- Segítségével hasonló eredményeket kaphatunk, mintha polinomikus jellemzőket adnánk az adathalmazhoz anélkül, hogy valójában hozzáadnánk őket.



Kernelizált SVM

- Tegyük fel, hogy másodrendű transzformációt alkalmazunk egy 2D tanító halmazra. Ekkor a másodrendű polinomikus leképező függvény φ (fí): ↓
- Látható, hogy a transzformált vektor 3 dimenziós!
- Ha ezt a másodrendű polinomikus leképezést alkalmazzuk, és kiszámoljuka vektorok belső szorzatait: ↓
- Vegyük észre, hogy a transzformált vektorok belső szorzata egyenlő az eredeti vektorok belső szorzatának négyzetével: $\phi(\mathbf{a})^T \phi(\mathbf{b}) = (\mathbf{a}^T \mathbf{b})^2$
- Kernelek:

$$\phi(\mathbf{x}) = \phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{pmatrix}$$

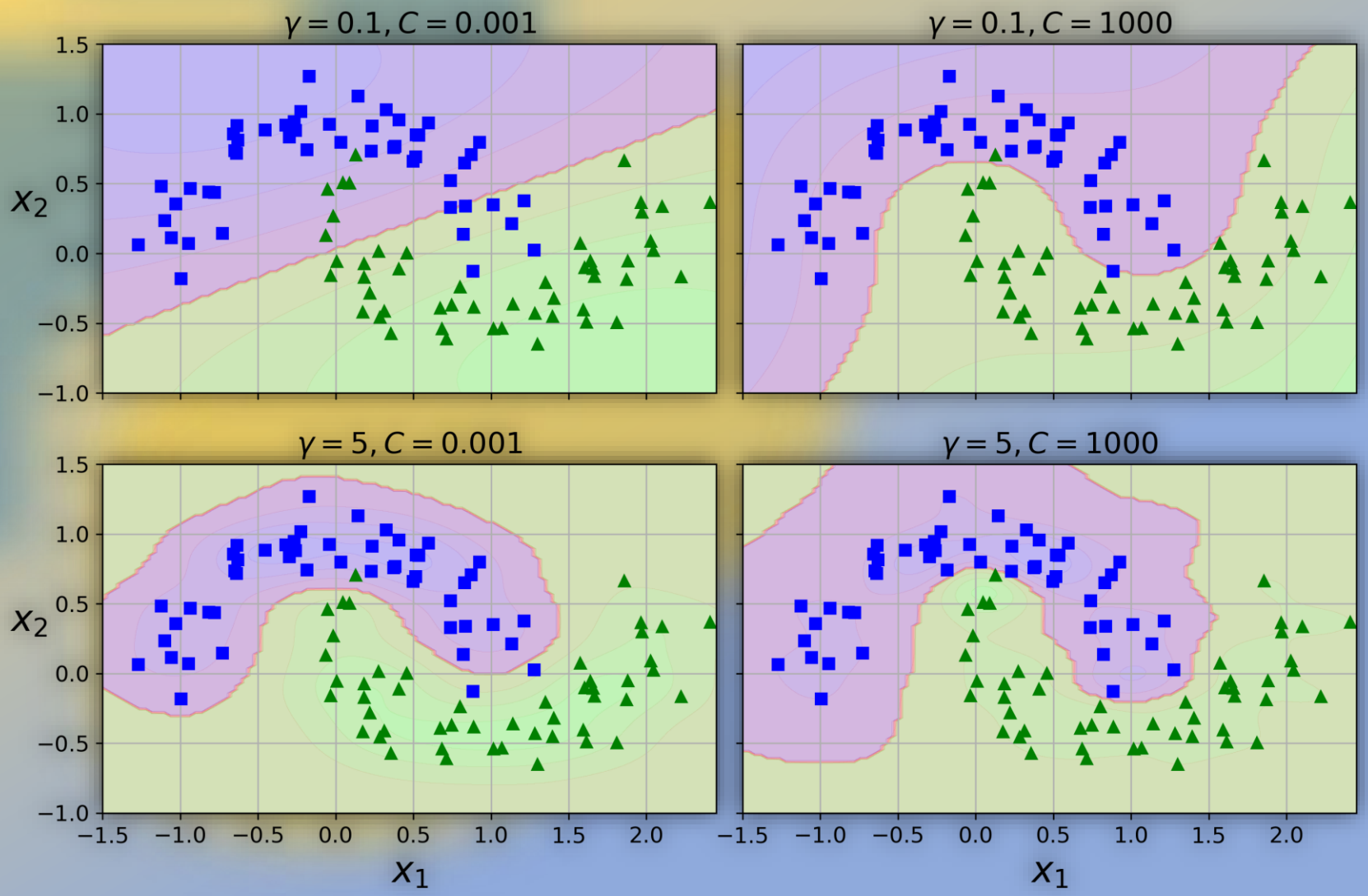
$$\phi(\mathbf{a})^T \phi(\mathbf{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}\, a_1 a_2 \\ a_2^2 \end{pmatrix}^T \begin{pmatrix} b_1^2 \\ \sqrt{2}\, b_1 b_2 \\ b_2^2 \end{pmatrix} = a_1^2 b_1^2 + 2 a_1 b_1 a_2 b_2 + a_2^2 b_2^2$$

$$= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right)^2 = (\mathbf{a}^T \mathbf{b})^2$$

Linear:	$K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b}$
Polynomial:	$K(\mathbf{a}, \mathbf{b}) = (\gamma \mathbf{a}^T \mathbf{b} + r)^d$
Gaussian RBF:	$K(\mathbf{a}, \mathbf{b}) = \exp(-\gamma \lVert \mathbf{a} - \mathbf{b} \rVert^2)$
Sigmoid:	$K(\mathbf{a}, \mathbf{b}) = \tanh(\gamma \mathbf{a}^T \mathbf{b} + r)$

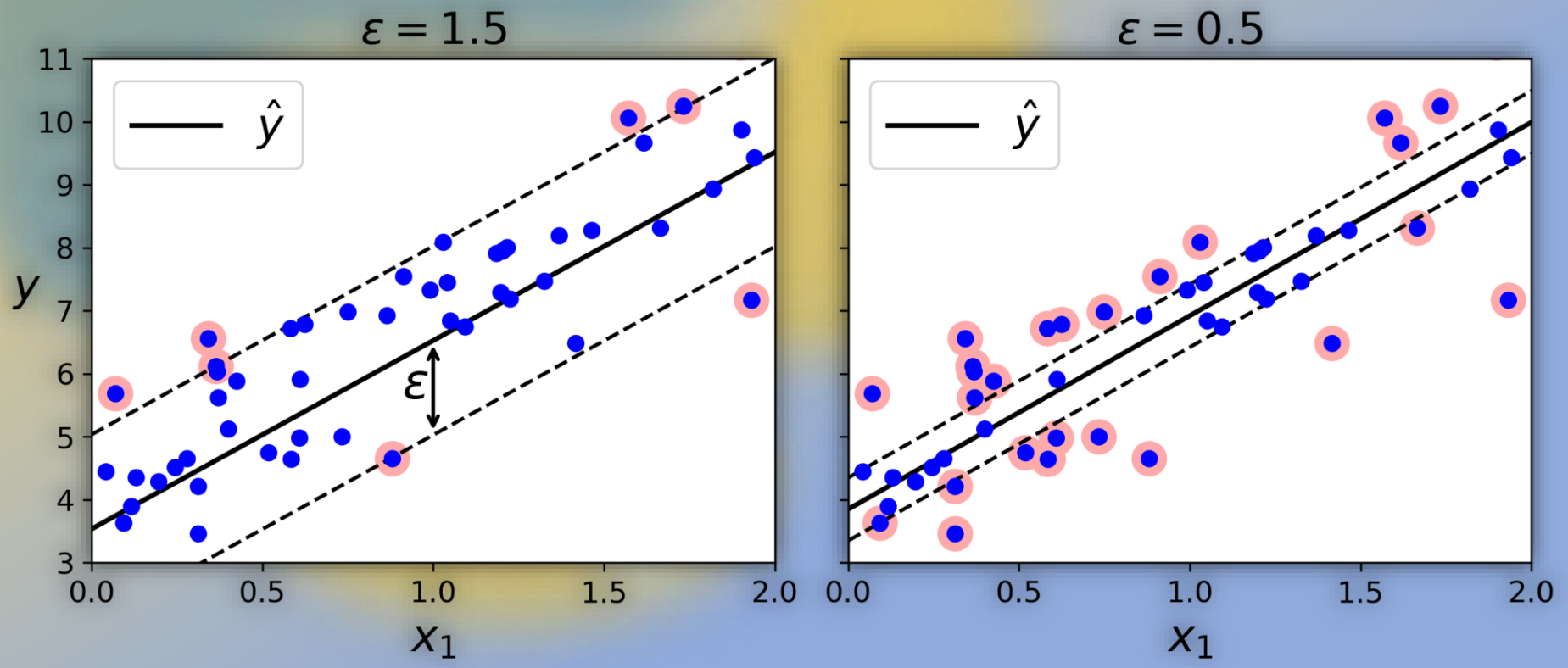
Radiális bázis függvények kernelként

- Ahogy a polinomikus jellemzők esetében, úgy a hasonlósági jellemzők esetén is működik a kernel trükk.
- A diagram különböző γ és C értékekkel tanított modelleket láthatunk.
- Figyeljük meg, ahogy a γ növelése szűkebb harang alakzatokat ad ereményül, ezért minden mintaegyed hasonlósági tartománya kisebb.
- A γ csökkentése pedig nagyobb befolyási teret adnak a pontoknak.



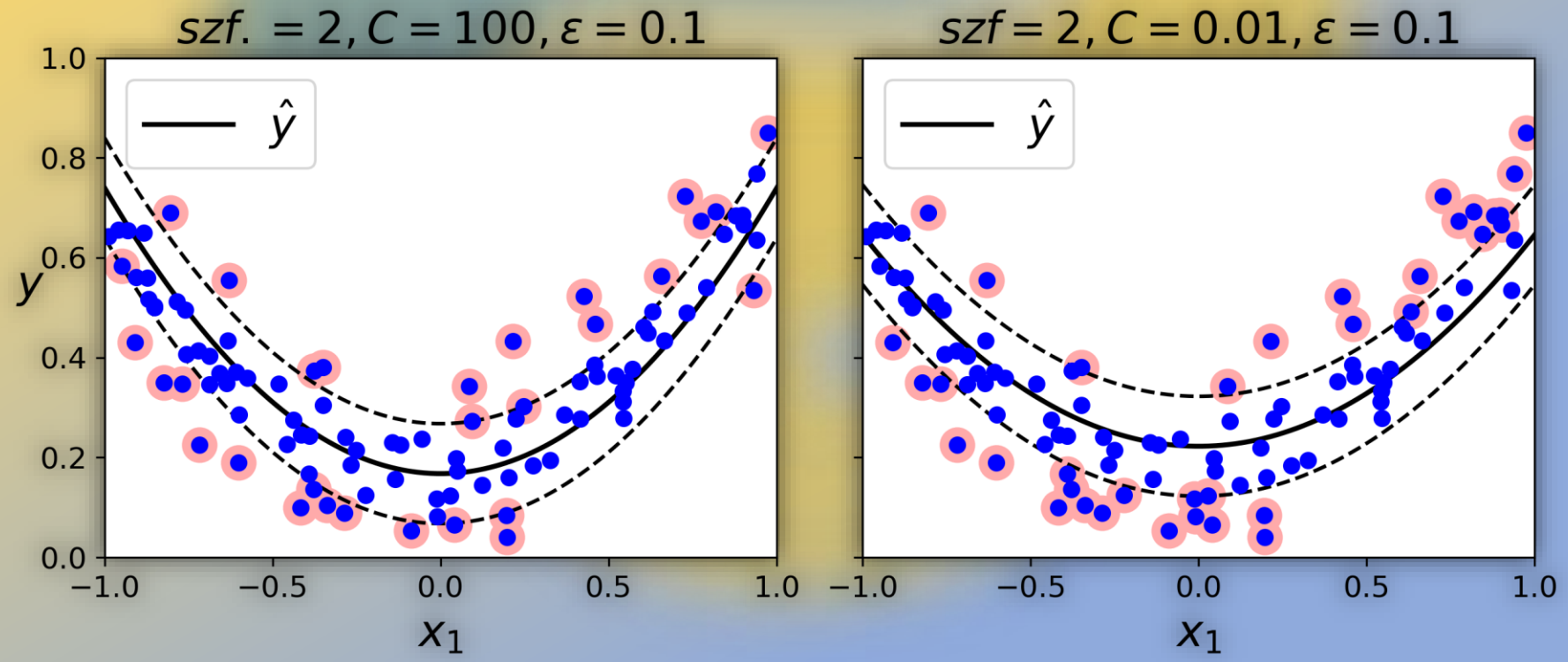
SVM regresszió

- Az SVM regresszió esetében a cél pont az osztályozás ellentéte: a legnagyobb út helyett az algoritmus megpróbálja a lehető legtöbb egyedet az útra ráhelyezni, a margósértések minimumon tartása mellett.
- Az út szélességét az ϵ (éta) szabályozza. Nagyobb $\epsilon \rightarrow$ nagyobb margó.
- A két diagramon két különböző ϵ értékkel tanított modellt láthatunk.
- ϵ-inszenzitivitás: a margónbelülre hozzáadott egyedek nem befolyásolják a modellpredikcióját.



Nemlineáris SVM regresszió

- A nemlineáris regressziós feladatok elvégzésére használhatunk kernelizált SVM regresszorokat.
- Az alábbi diagramokon kvadratikus adatokon másodfokú polinomikus kernellel tanított SVM regresszorokat láthatunk.
- A bal oldali ábrán sok, míg a jobb oldalin kevés regularizáció játszik szerepet.

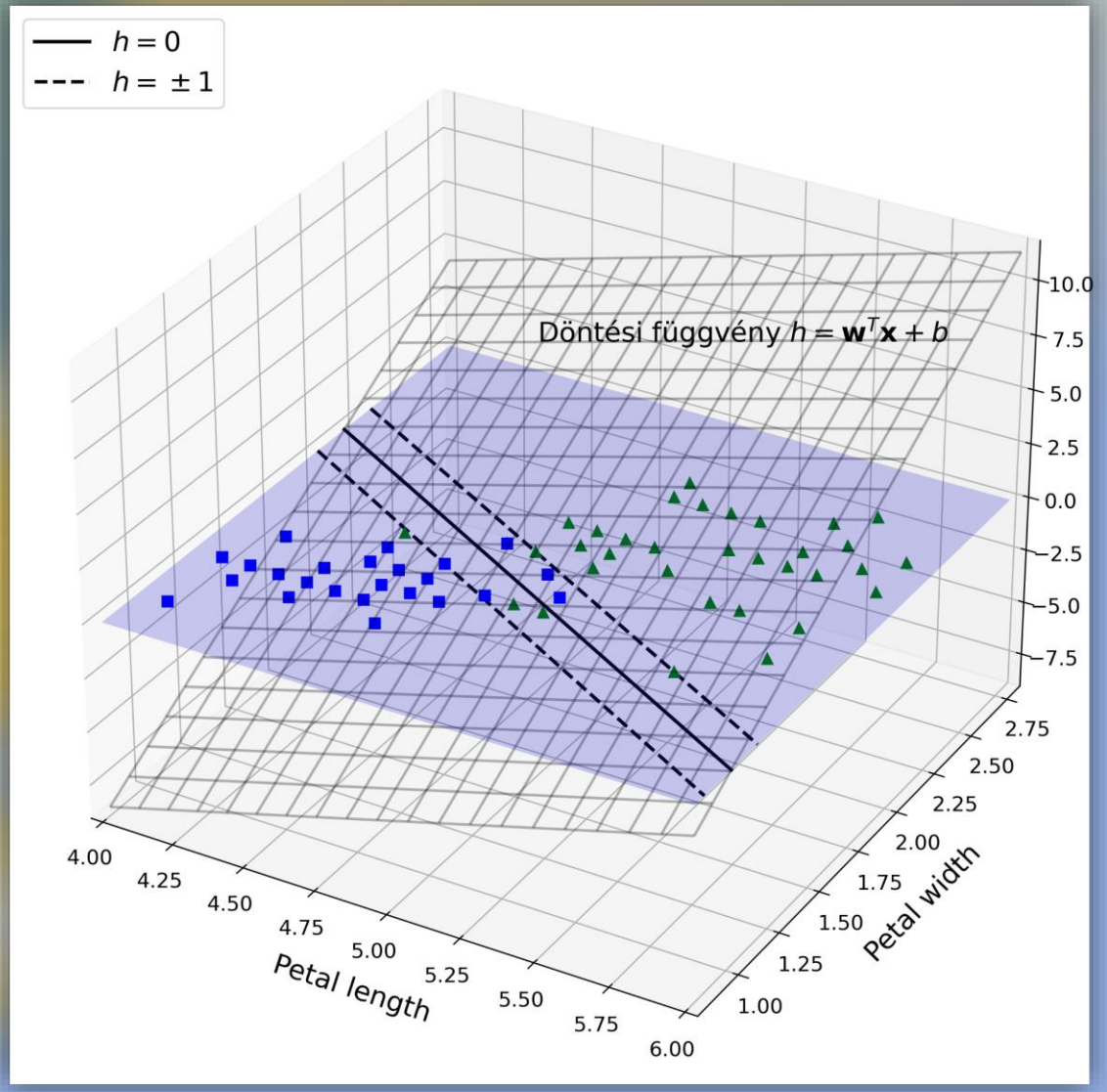


Az SVM működésének részletei

- A lineáris SVM osztályozó úgy állítja elő a predikciókat, hogy kiszámolja a döntési hipersík képletét:
  $w^T x + b = w_1 x_1 + \cdots + w_n x_n + b.$
- Ha az eredmény pozitív, a predikció ($\hat{y}$) a pozitív osztály (1), minden más esetben pedig a negatív osztály (0).

$$\hat{y} = \begin{cases} 0 \text{ if } \mathbf{w}^T\mathbf{x} + b < 0, \\ 1 \text{ if } \mathbf{w}^T\mathbf{x} + b \geq 0 \end{cases}$$

- Az ábrán az 5. oldal bal oldali modelljének megfelelő döntési hipersíkot látjuk.
- A döntési határ azon pontok halmaza, ahol a döntési függvény 0.

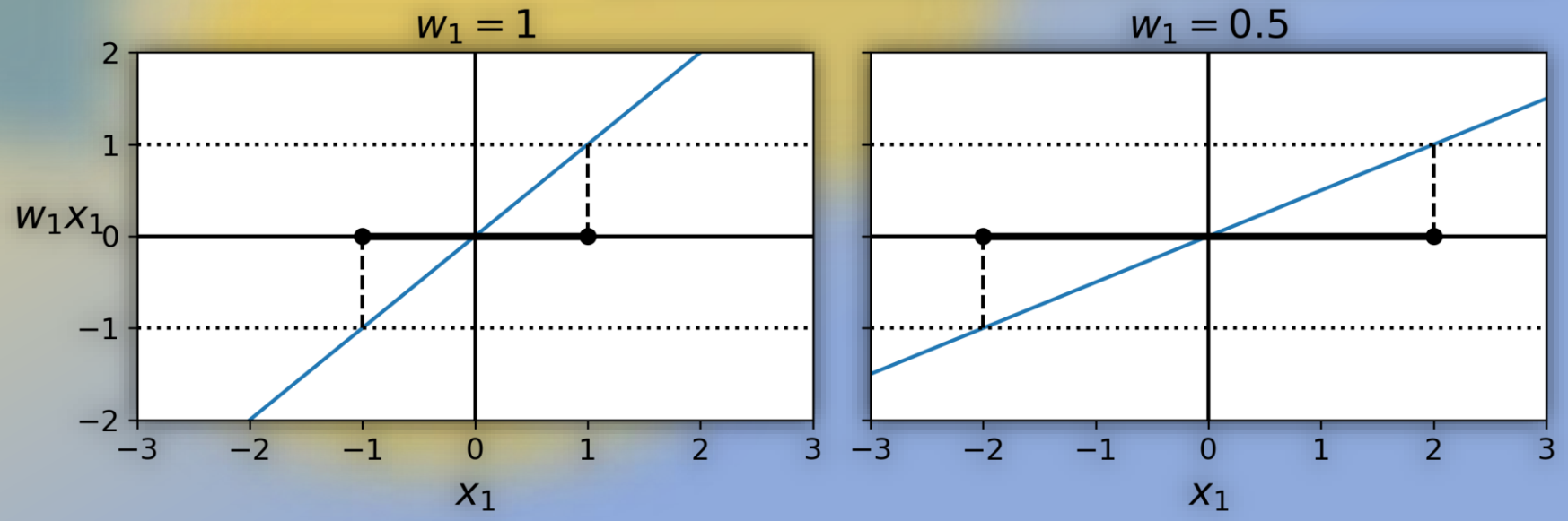


A Keménymargós SVM tanításának célfüggvénye

- Vegyük a döntési függvény meredekségét: ez egyenlő a súlyvektor normáltjával: $\|w\|$. A meredekség értéke fordítottan arányos a margó méretével: nagyobb meredekség kisebb margót eredményez.
- Ezt a $\|w\|$-t szeretnénk minimalizálni, hogy nagy margót kapjunk.
- Ha szeretnénk kikötni, hogy ne legyenek margósértések, a döntési függvénynek a pozitív egyedek esetén 1-nél nagyobbnak, a negatív egyedek esetén pedig -1-nél kisebbnek kell lennie.

$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad t^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right) \geq 1 \quad \text{for } i = 1, 2, \cdots, m$$



A lágymargós SVM tanításának célfüggvénye

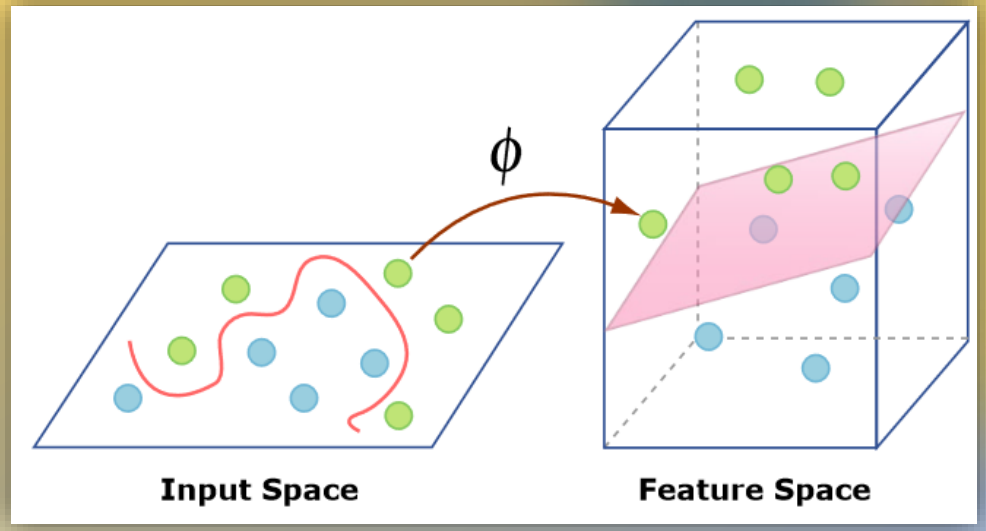
- A lágymargós célfüggvényhez bevezetünk egy kiegyenlítő változót ($\zeta^{(i)} > 0$), ami azt jelenti, hogy egy adott egyed milyen mértékben sértheti a margót.
- Így két egymással konfliktusban álló célunk van: a kiegyenlítő változót minimalizálni a margósértések miatt, és az $\frac{1}{2} w^T w$ kifejezést minimalizálni a margó növelése végett.
- Itt jön be a C hiperparaméter: megengedi, hogy definiáljuk az egyensúlyt a két célváltozó között. Ez a következő megkötött optimalizálási problémához vezet:

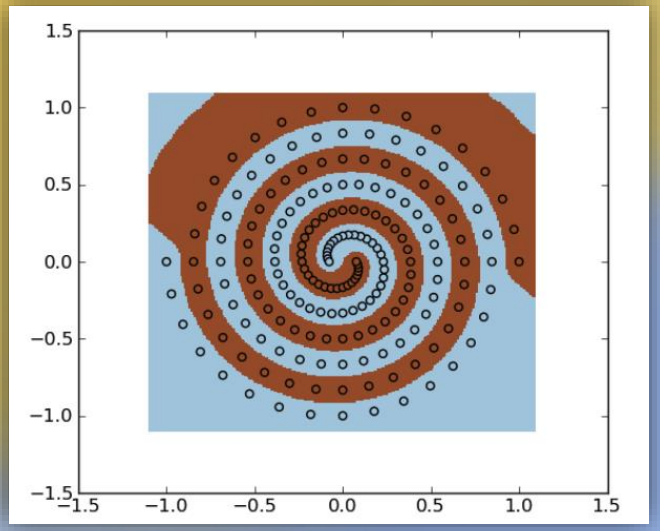
$$\underset{\mathbf{w}, b, \zeta}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\zeta^{(i)}$$

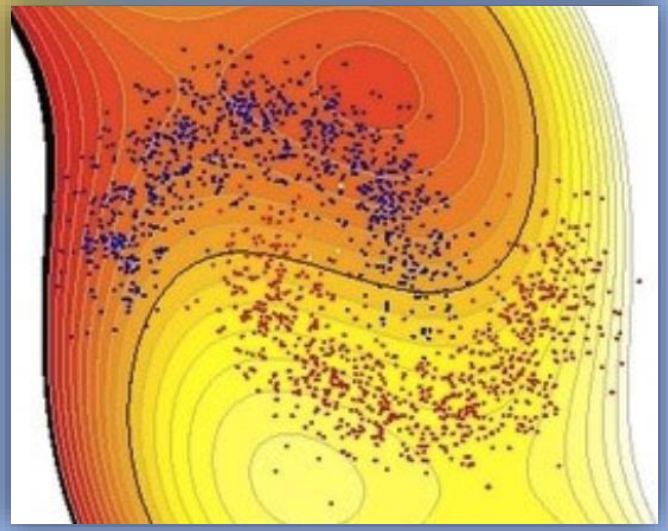
$$\text{subject to} \quad t^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right) \geq 1 - \zeta^{(i)} \quad \text{and} \quad \zeta^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$

Mercer tézise

- A tézis szerint, ha egy függvény $K(a, b)$ teljesít egy pár matematikai kitételt (folytonos, szimmetrikus argumentumok szerint: $K(a, b) = K(b, a)$), akkor létezik egy olyan φ leképezés, amely a-t és b-t egy olyan, sokkal magasabb dimenziós térbe képezi le, amelyre igaz, hogy $\mathrm{K}(a, b) = \varphi(a)^T \varphi(b)$.
- Tehát lehet használni K-t kernelként, ha nem is tudjuk biztosan, hogy φ pontosan mi is. Elég, ha tudjuk, hogy φ létezik.
- A Gauss-i RBF kernel esetén a leképezés egy végtelen-dimenziós térbe történik.







Egy híres példa

- Egy terrorista kirakta magáról a bal oldali képet, azzal a szöveggel, hogy „kapjatok el ha tudtok”.
- A CIA erre egyszerűen „lecsavarta” a kép spirálját, láthatóvá téve az arcát. A műveletben egy vektorgépet használtak.