

SVM

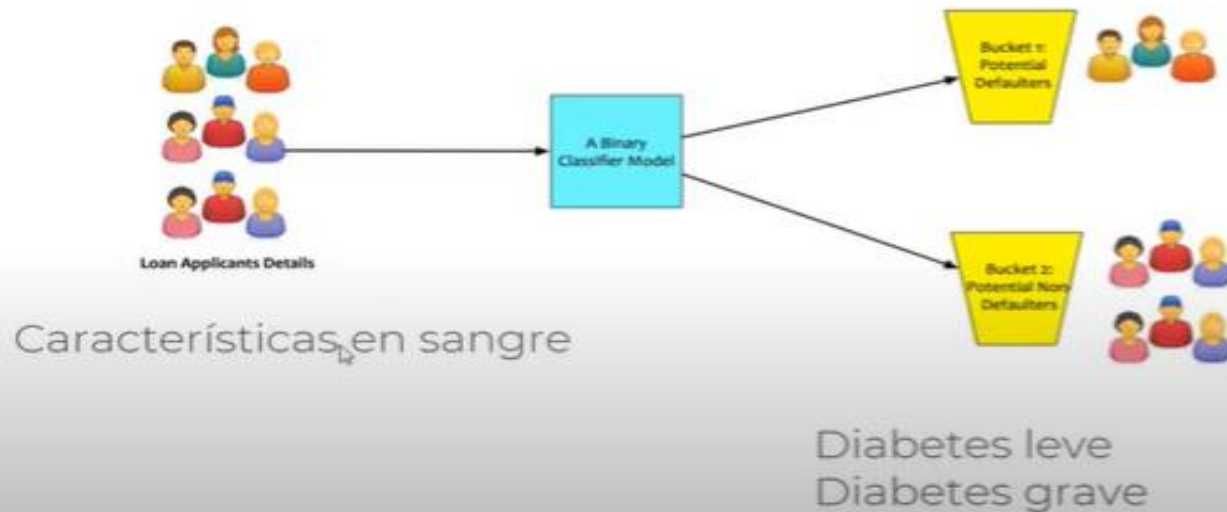
Machine Learning con R

SVM Como clasificador



SVM Como clasificador

Clasificación (supervisado)



Objetivo:

A partir de una serie de variables predictoras (cuantitativas o cualitativas) poder predecir el grupo de pertenencia

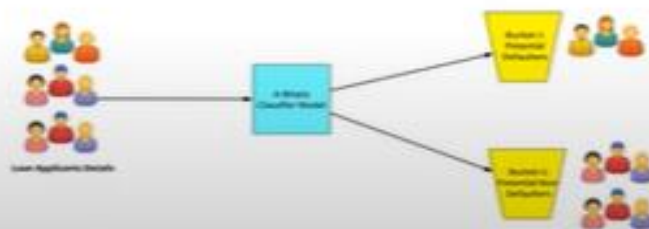
“A partir de características en sangre poder predecir si tenemos riesgo bajo o alto de padecer diabetes”

SVM Como clasificador

Esquema de la clasificación



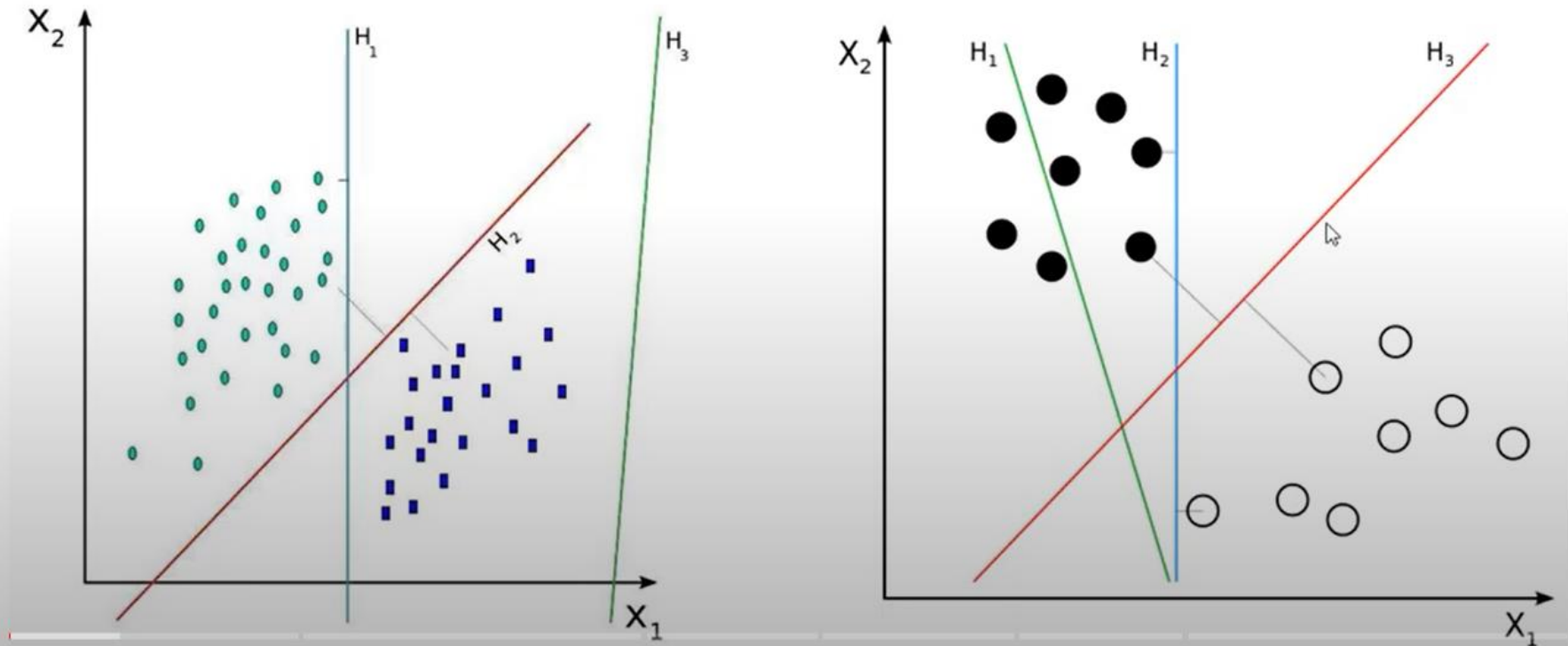
Peso
Edad
Nivel de glucosa medio
...



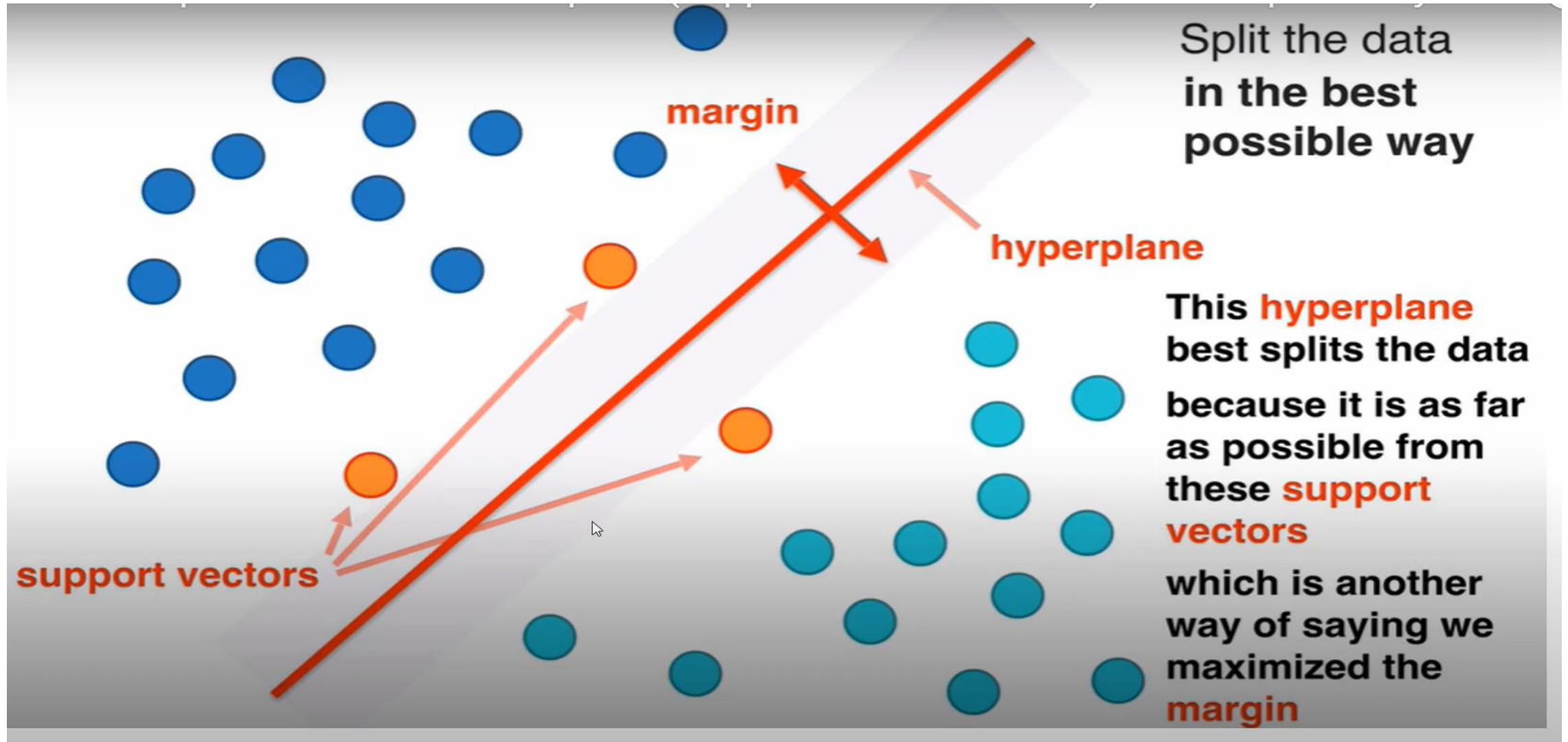
Diabetes:
Leve
Grave

SVM Como clasificador

- Trata de calcular vectores (direcciones) de rectas que mejor separen a los grupos
- La recta (hiperplano) con un margen más grande a los puntos de clases será la ganadora



SVM Como clasificador

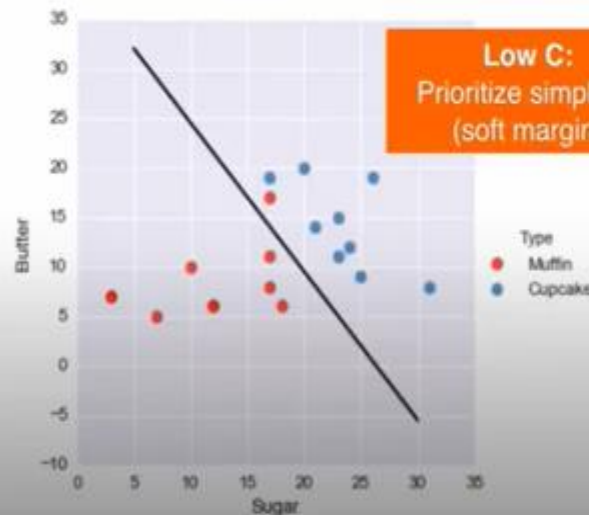


SVM Como clasificador

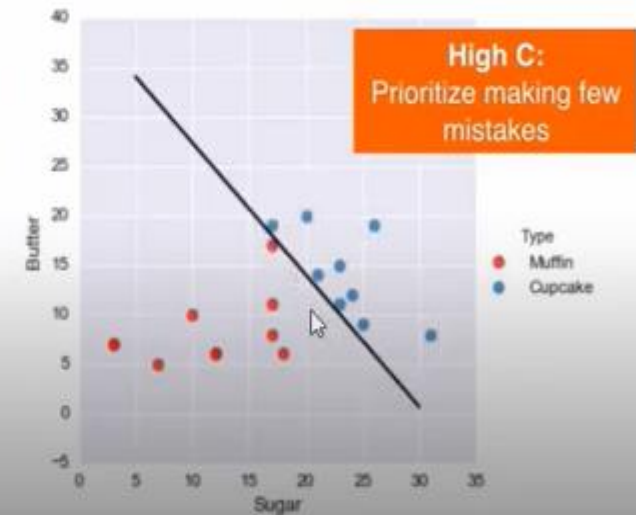
- Idealmente, el modelo basado en SVM debería producir un hiperplano que separe completamente los datos del universo estudiado en dos categorías. Sin embargo, una separación perfecta no siempre es posible y, si lo es, el resultado del modelo no puede ser generalizado para otros datos. Esto se conoce como sobreajuste (overfitting).
- Con el fin de permitir cierta flexibilidad, las SVM manejan un parámetro C que controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando (soft margin) que permita algunos errores en la clasificación a la vez que los penaliza.

C Parameter: Comparison

```
# Fit the SVM model with a LOW C  
model = svm.SVC(kernel='linear', C=2**-5)  
model.fit(sugar_butter, type_label)
```



```
# Fit the SVM model with a HIGH C  
model = svm.SVC(kernel='linear', C=2**5)  
model.fit(sugar_butter, type_label)
```



Higher Dimensions

C Parameter

Multiple Classes

Kernel Trick

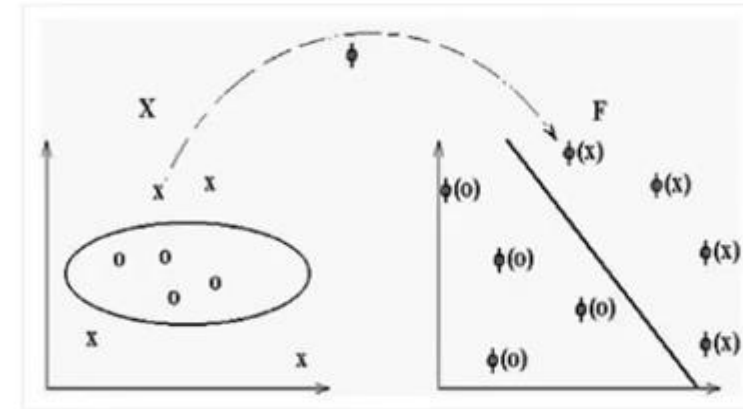
SVM Como clasificador

Casos reales – Kernel Trick

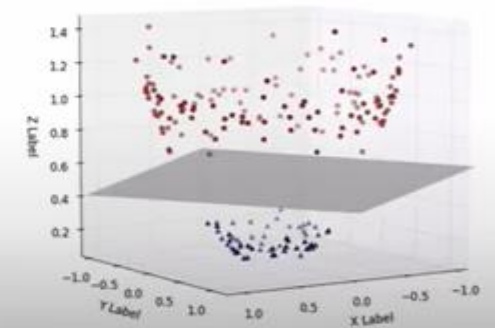
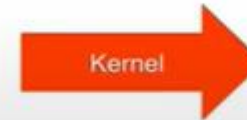
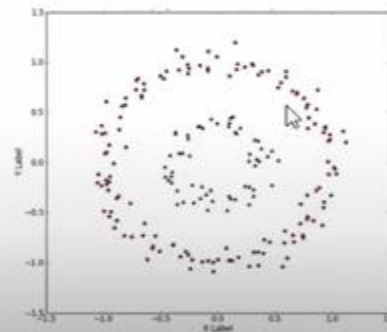
- La manera más simple de realizar la separación es mediante una línea recta, un plano recto o un hiperplano N-dimensional.
- Desafortunadamente los universos a estudiar no se suelen presentar en casos idílicos de dos dimensiones como en el ejemplo anterior, sino que un algoritmo SVM debe tratar con a) más de dos variables predictoras, b) curvas no lineales de separación, c) casos donde los conjuntos de datos no pueden ser completamente separados, d) clasificaciones en más de dos categorías.
- Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquinas de aprendizaje lineal. Es decir, mapearemos el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad (Hilbert):

$$F = \{\phi(x) | x \in X\}$$

$$x = \{x_1, x_2, \dots, x_n\} \rightarrow \phi(x) = \{\phi_1(x), \phi_2(x), \dots, \phi_n(x)\}$$



Kernel Trick: Visual



Kernel Options

- Linear
- Radial Basis Function
- Polynomial
- Sigmoid

Higher Dimensions

C Parameter

Multiple Classes

Kernel Trick

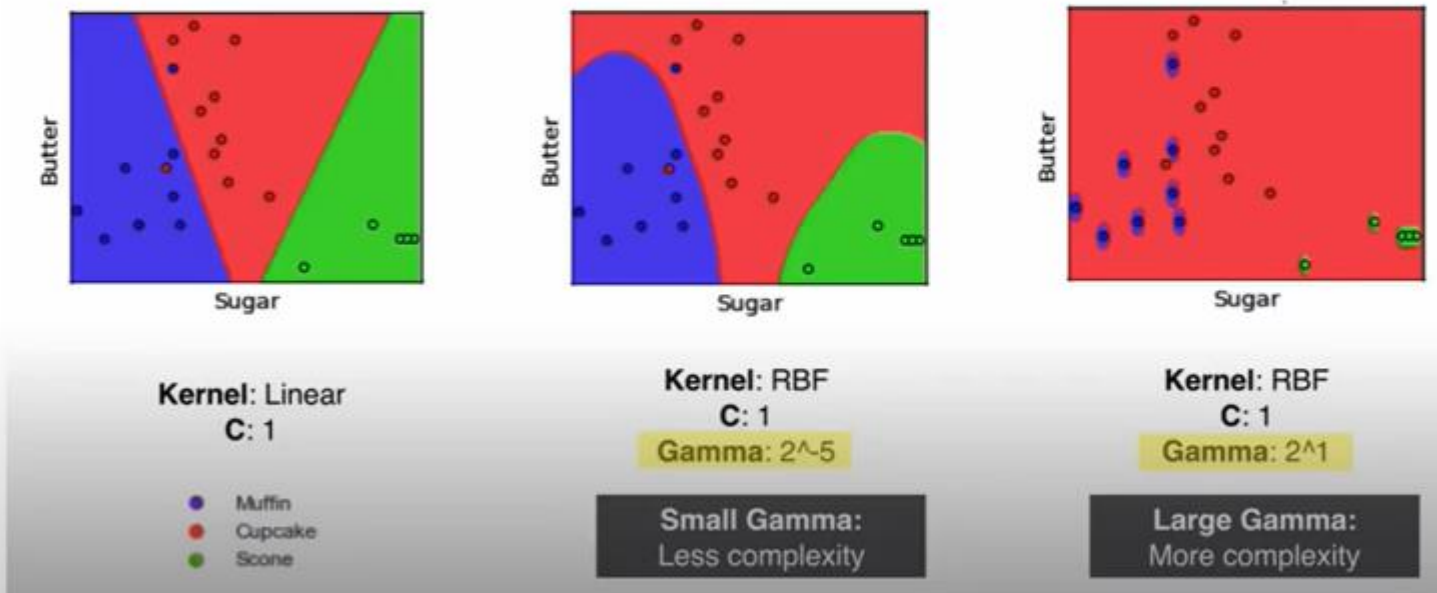
SVM Como clasificador

Casos reales – Radial Kernel vs Lineal Kernel

- Una de las funciones Kernel más útiles es la funciones radiales que lleva a un espacio infinito dimensional
- Función de base radial Gaussiana: separado por un hiperplano en el espacio transformado.

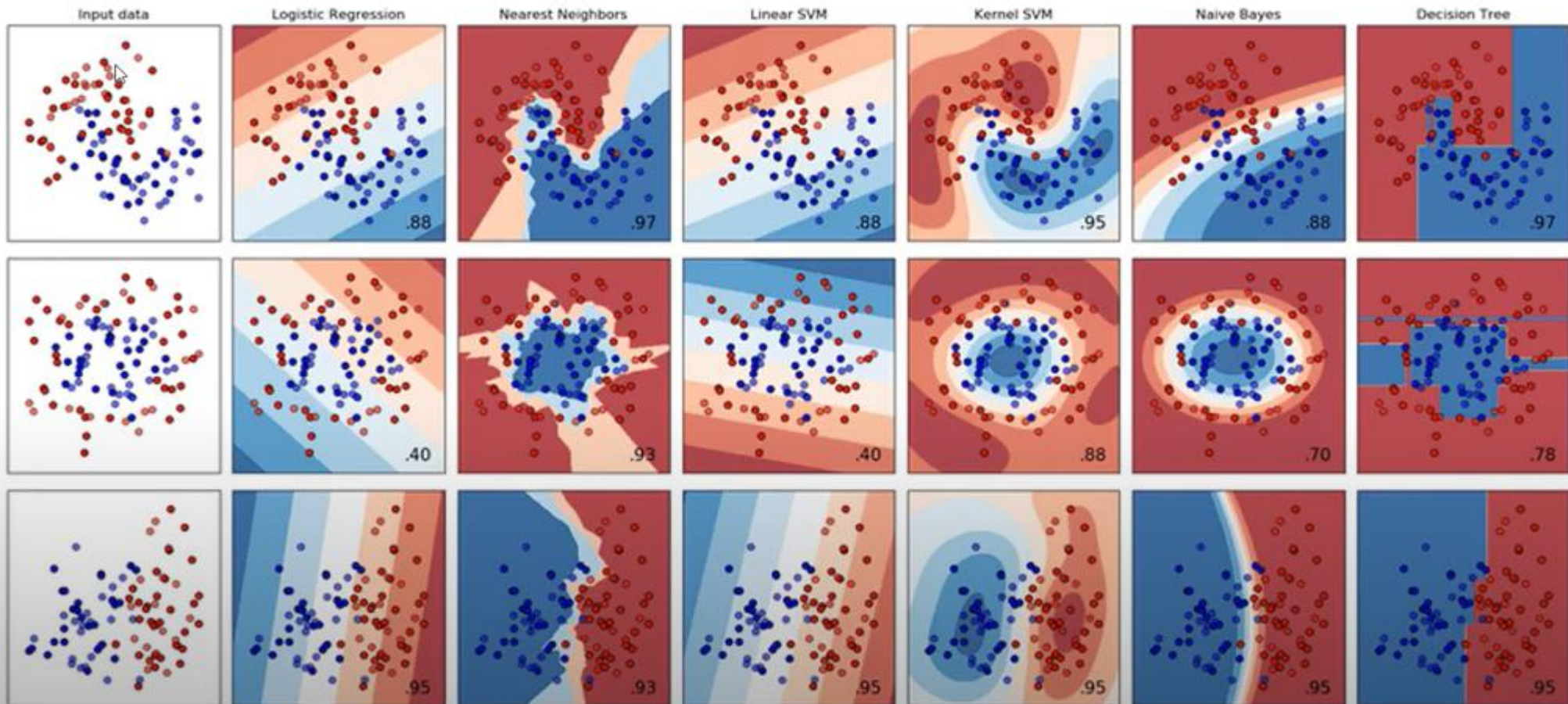
$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

Kernel Trick: Comparison



SVM Como clasificador

Otras funciones Kernel y otros algoritmos



Librerías en R

- `install.packages("kernlab")`
- `library(kernlab)`
- `modelo <- ksvm()`

- `install.packages("e1031")`
- `library(e1031)`
- `modelo <- train()`
- <https://rpubs.com/rpizarro/605730>

Bibliografía

- <https://www.youtube.com/watch?v=QoRBenaGzzw>
- <https://rpubs.com/rpizarro/605730>