



Universidad Autónoma de Nuevo León

**Facultad de ciencia Físico Matemáticas
Minería de Datos**

Máquina de soporte vectorial (SVM)

Grupo: 003

Equipo 5:

Edson Alí González Campos

Francisco Everardo López Domínguez

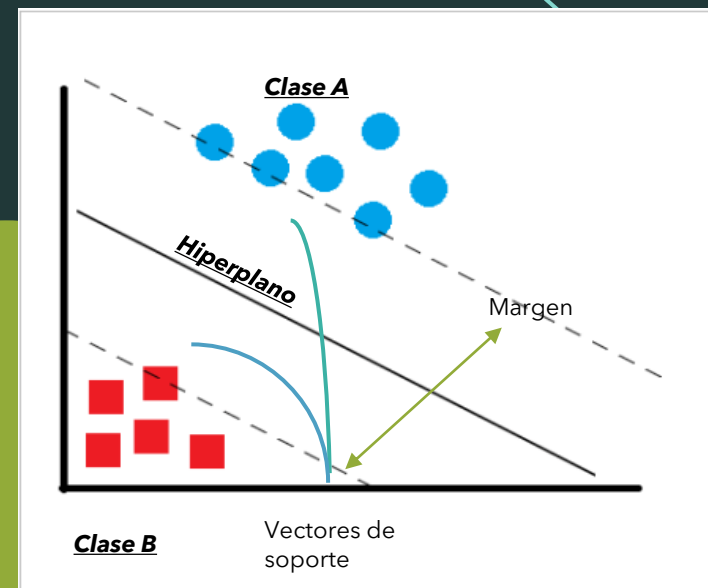
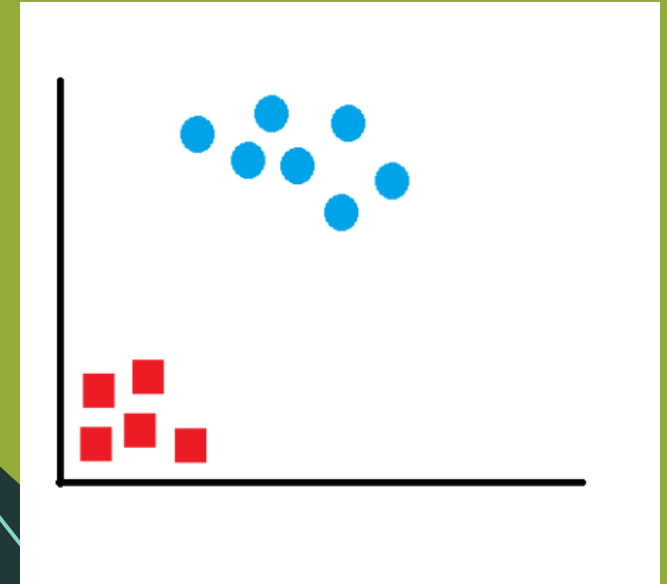
Anakaren Segovia González

Jennifer Jacqueline Herrera Rivera

¿Qué es?

Son un conjunto de algoritmos de aprendizaje, supervisado y desarrollados por Vladimir Vapnik y su equipo en 1992, usados para resolver problemas de clasificación y regresión.

Son algoritmos que aplicados a una serie de datos, donde se clasifican estableciendo un espacio entre ellos y una línea ya sea curva o recta, de forma que cualquier punto predicho estará en uno de los dos lados o regiones que genera dicha línea en el espacio.



Usos:

- Clasificación binaria (aplicación original)
- Clasificación multiclase
- Regresión
- Selección de variables
- Identificación de casos anómalos
- Clustering

Características:

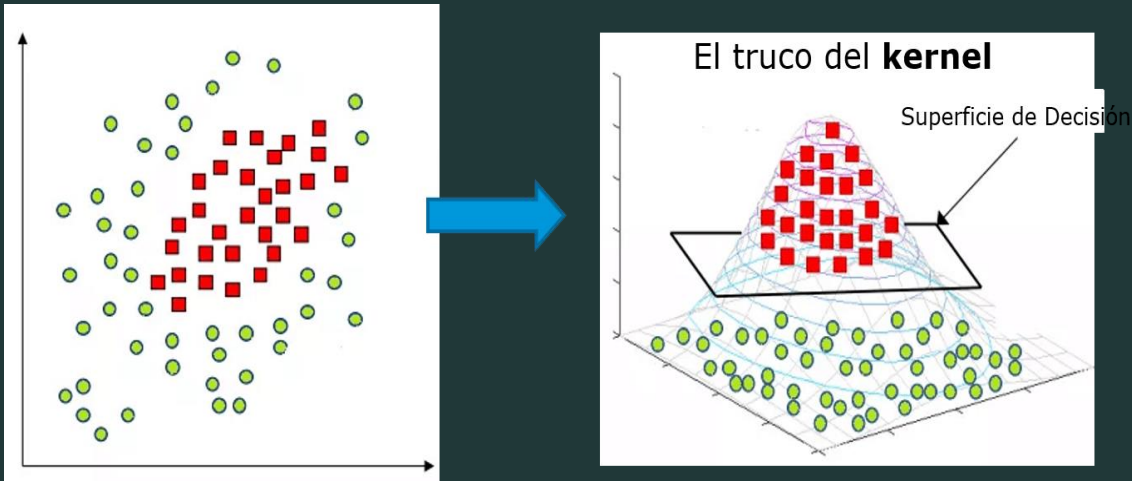
- Técnica de clasificación de datos
- Más fácil de utilizar que las redes neuronales
- Trabaja en un espacio de características inducido por la función Kernel

Objetivos de las SVM:

- Dado un conjunto de elementos de entrenamiento (de muestras) queremos etiquetar los datos en 2 categorías en una base de una línea, y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra futura.

Representación por medio de funciones núcleo ó Kernel

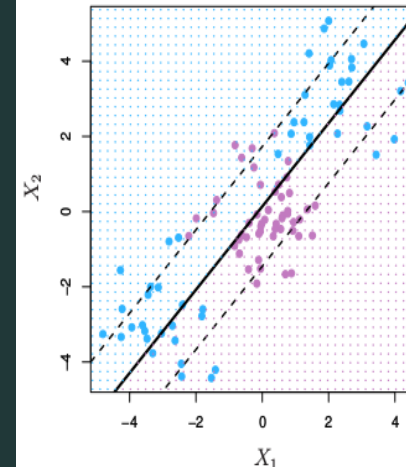
Ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión, el cual aumenta la capacidad computacional de la máquinas de aprendizaje lineal.



Funciones del Kernel

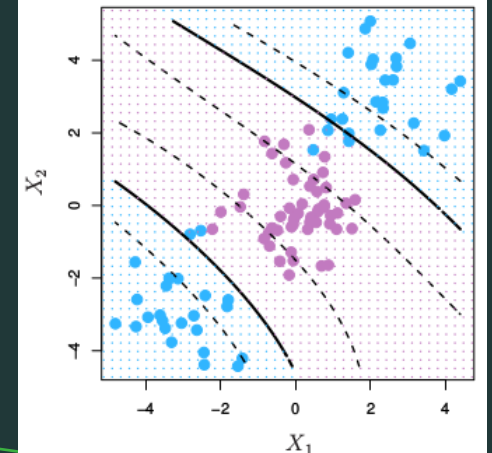
Kernel lineal

$$K(x, x') = x \cdot x'$$



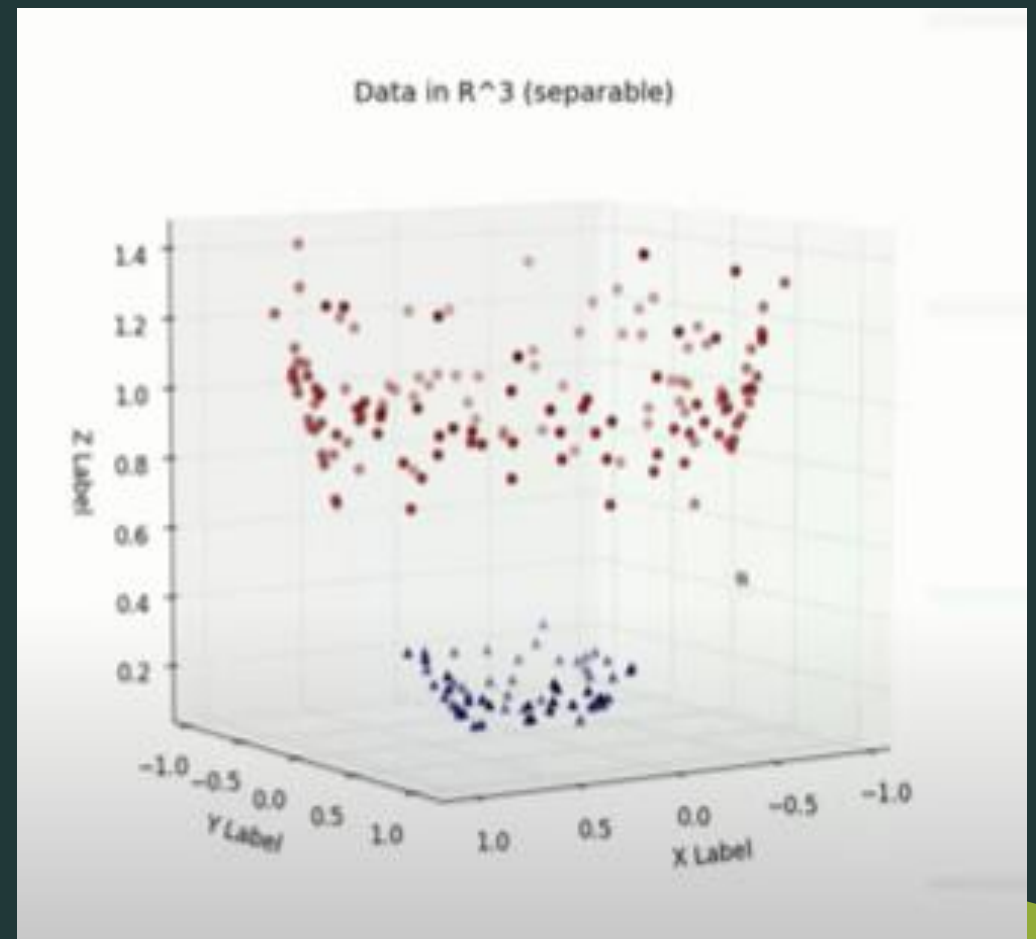
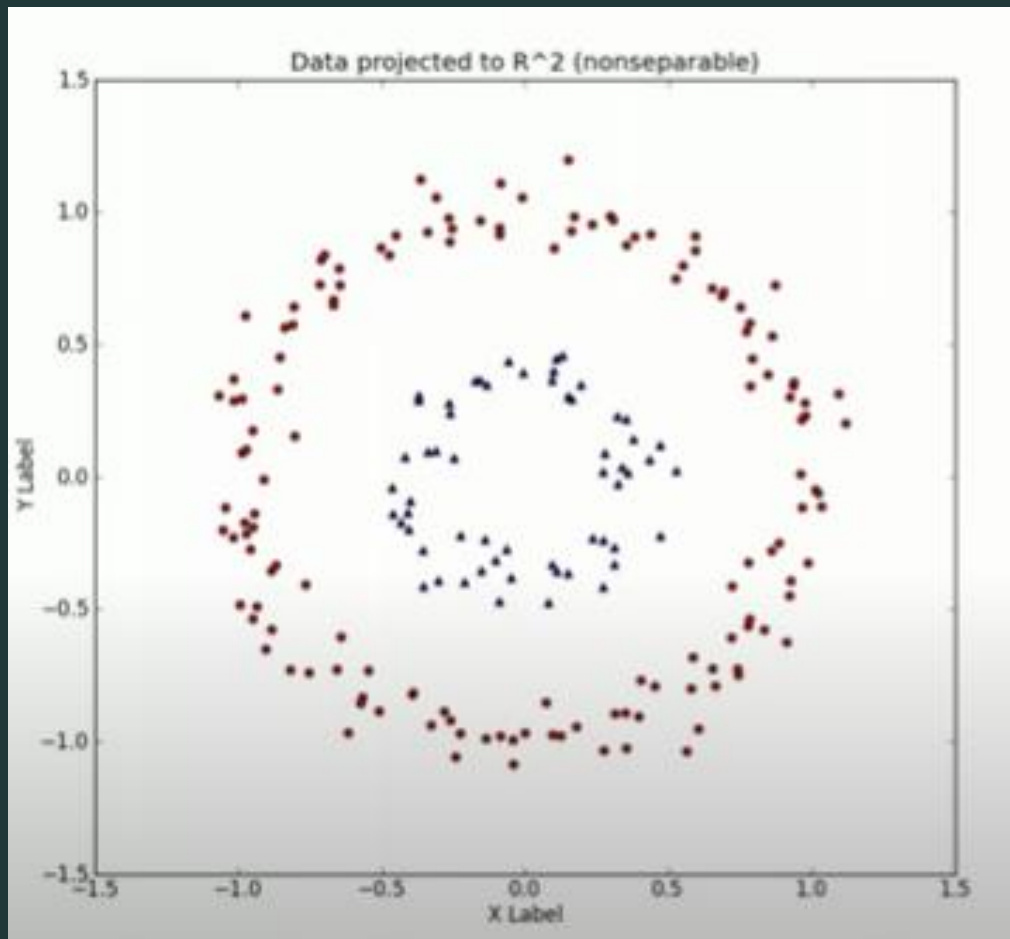
Kernel polinómico

$$K(x, x') = (x \cdot x' + c)^d$$



Ejemplo:

- Este ejemplo lo vamos a representar como vectores en un espacio m -dimensional (m = numero de atributos)



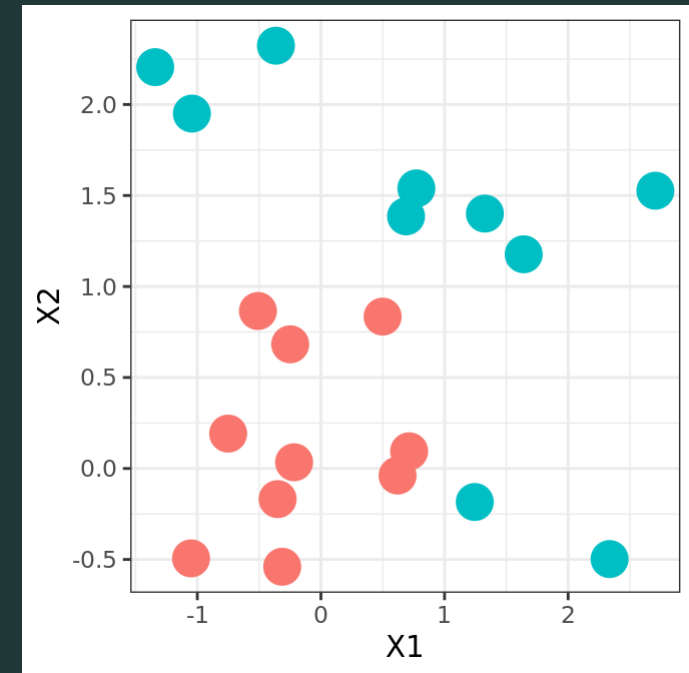
• EJERCICIO

- Para mostrar el uso de un *Support Vector Classifier* como clasificador binario, se simulan observaciones en un espacio bidimensional que pertenecen a dos clases. Este ejemplo se ha obtenido de los videos asociados al libro *Introduction to Statistical Learning*, que no es igual al presentado en el libro.
- En los siguientes ejemplos, se emplea la función `svm()` contenida en el paquete `e1071`. Esta función ajusta *Support Vector Classifier* si se le especifica el argumento `kernel="linear"` (como se describe más adelante, el método de *Support Vector Machines* es equivalente al *Support Vector Classifier* cuando el kernel utilizado es lineal). El argumento `cost` determina la penalización aplicada por violar el margen, es el nombre que emplea esta función para el hiperparámetro $\gamma(C)$.

```
set.seed(10111)
coordenadas <- matrix(rnorm(40), 20, 2)
colnames(coordenadas) <- c("X1", "X2")
y <- c(rep(-1,10), rep(1,10))
coordenadas[y == 1, ] <- coordenadas[y == 1, ] + 1
datos <- data.frame(coordenadas, y)
ggplot(data = datos, aes(x = X1, y = X2, color = as.factor(y))) +
  geom_point(size = 6) +
  theme_bw() +
  theme(legend.position = "none")
```

La representación gráfica de los datos muestra que los grupos no son linealmente separables.

La función `svm()` identifica automáticamente si se trata de un problema de clasificación, la variable respuesta es de tipo factor, o de regresión, la variable respuesta es tipo numérico.



```
library(e1071)
```

```
# Se convierte la variable respuesta a factor  
datos$y <- as.factor(datos$y)
```

```
# Para que la función svm() calcule el Support Vector Classifier,  
# se tiene que indicar que la función kernel es lineal.  
modelo_svm <- svm(formula = y ~ X1 + X2, data = datos, kernel = "linear",  
                  cost = 10, scale = FALSE)
```

IMPORTANTE: En este caso, ambos predictores (X1, X2) tienen la misma escala por lo que no es necesario estandarizarlos. En aquellas situaciones en las que las escalas son distintas, sí hay que estandarizarlos, de lo contrario, los predictores de mayor magnitud eclipsarán a los de menor magnitud.