cluster 2018

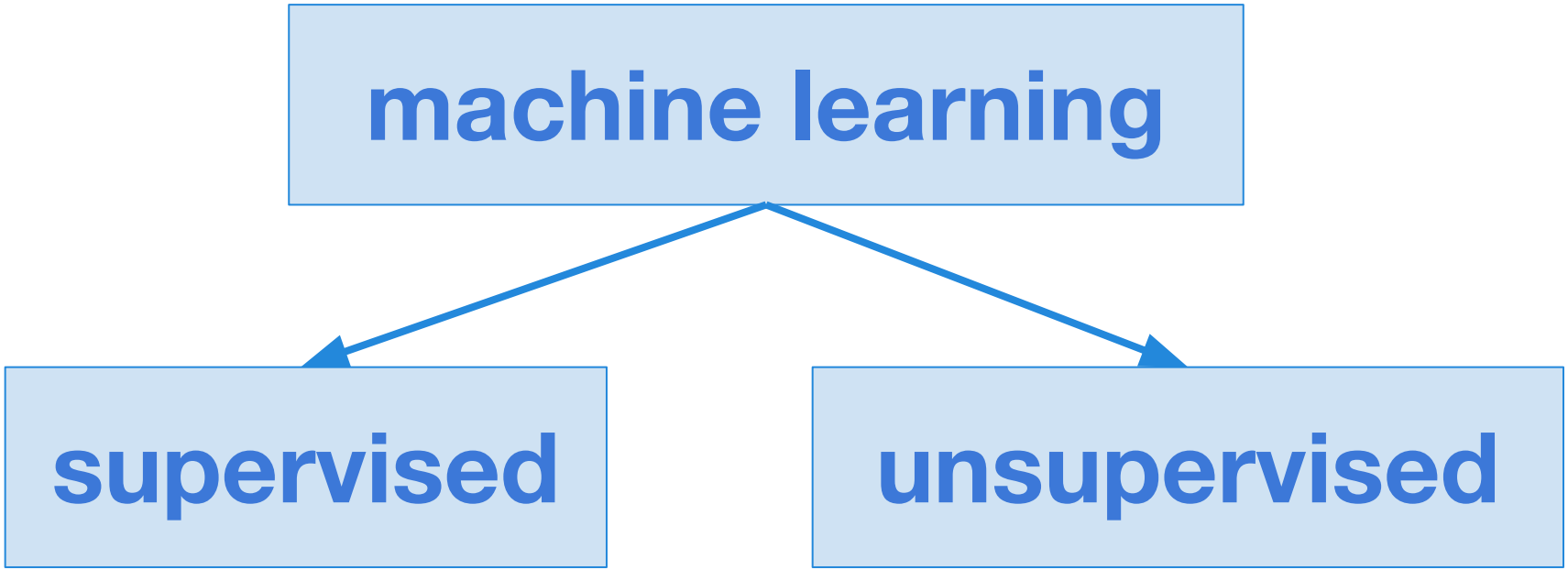
ciencia de datos en ingeniería industrial

clase_03: clasificación

agenda clase03: aprendizaje supervisado

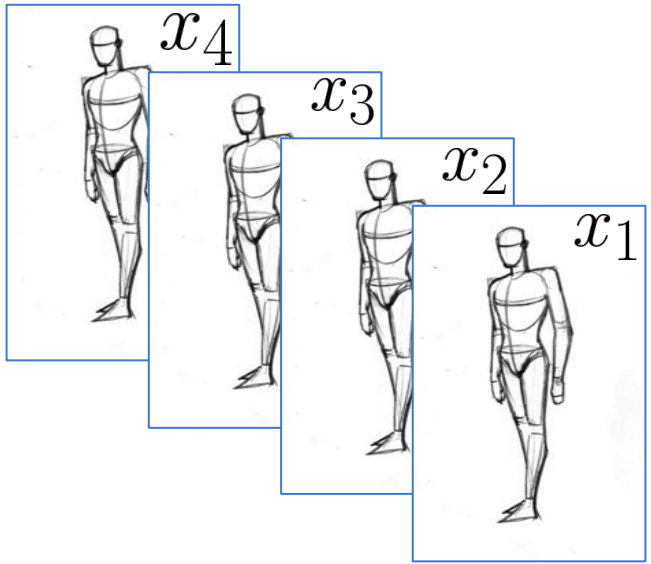
- Clasificación, binary class, multiclass
- Train, validation, test
- Cross validation
- Grid Search
- Confusion Matrix
- Performance metrics (Sens, Spec, ROC)
- Variance-bias trade off, overfitting
- Learning curve
- SVM
- KNN
- Logistic Regression

learning approaches



Samples and Features

$$X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, ..., x_{in})$$



X1**1** (feat. 1 in sample 1)

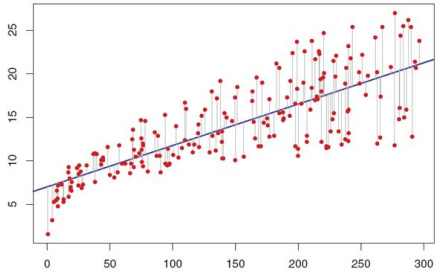
X1**2** (feat. 2 in sample 1)

X1**3** (feat. 3 in sample 1)

X1**4** (feat. 4 in sample 1)

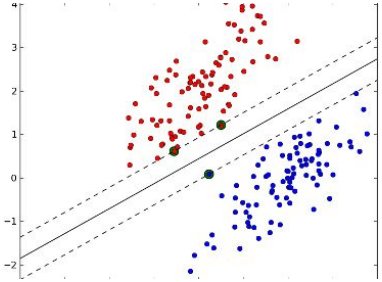
supervised learning methods

regression →



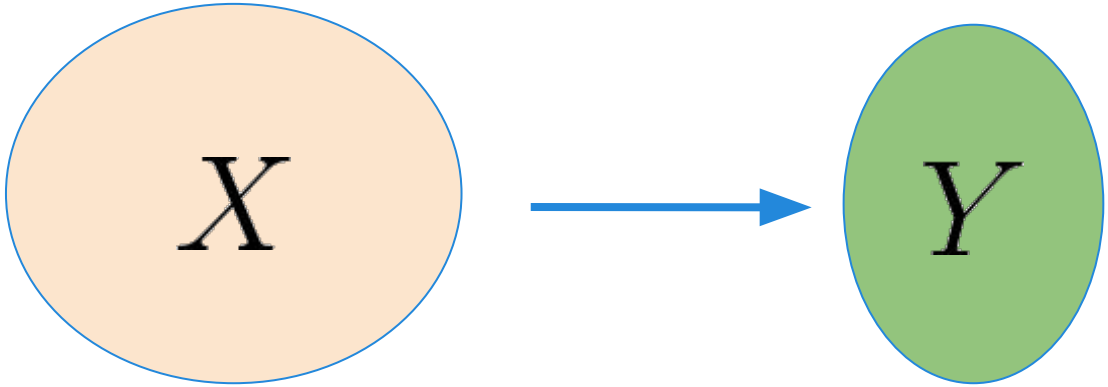
Y is continual

classification →



Y is categorical

supervised learning



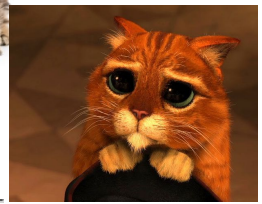
- We assume Y variable depends on X.
- **What we do not know is the true function/rule y = f(x).**

supervised learning: classification

pic01 X_1

pic02 X_2

pic03 X_3



pic04 X_4

DOG y_1

DOG y_2

CAT y_3

CAT y_4

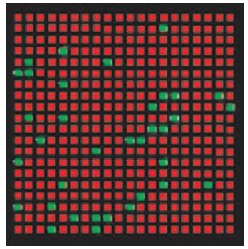
Data type: images
Features: Pixels

Each sample has associated a label settled by a human user.

supervised learning: classification

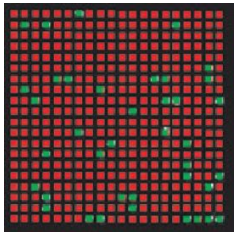
Data type: microarrays
Features: Genes

patient01 X_1



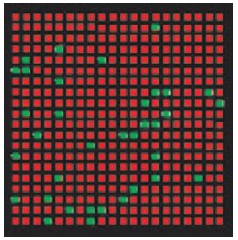
Healthy Tissue y_1

patient02 X_2



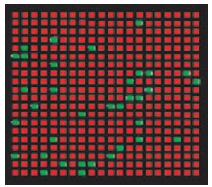
Cancer Tissue y_2

patient03 X_3



Cancer Tissue y_3

supervised learning: classification

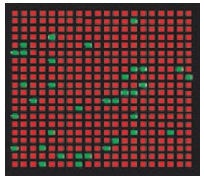


sample01 X_1

Healthy Tissue y_1

sample02 X_2

Cancer Tissue y_2

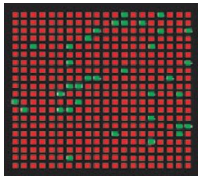


sample03 X_3

Cancer Tissue y_3

Trained Model

Prediction

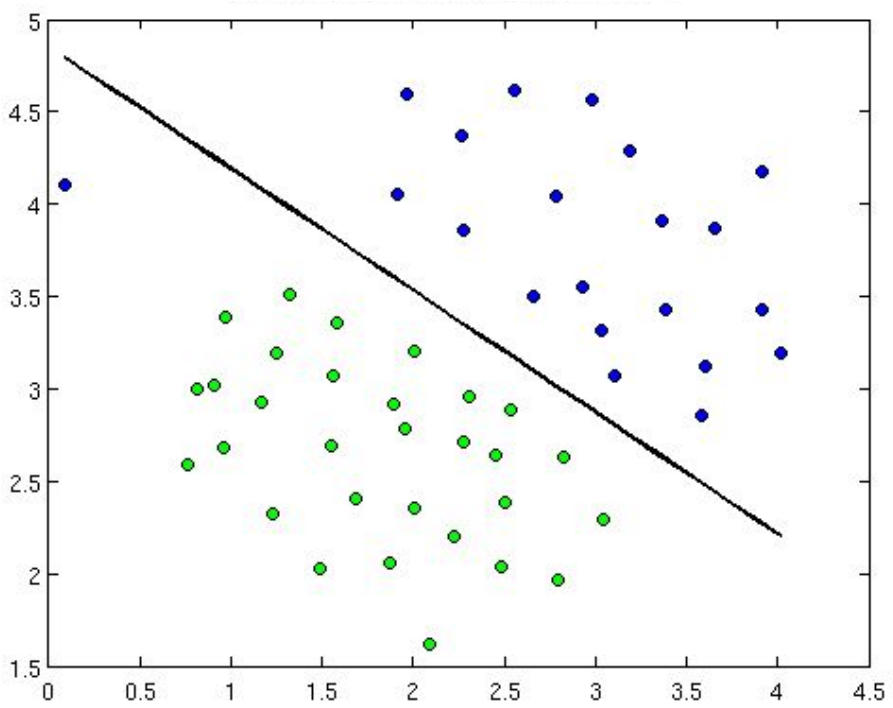


New sample

????

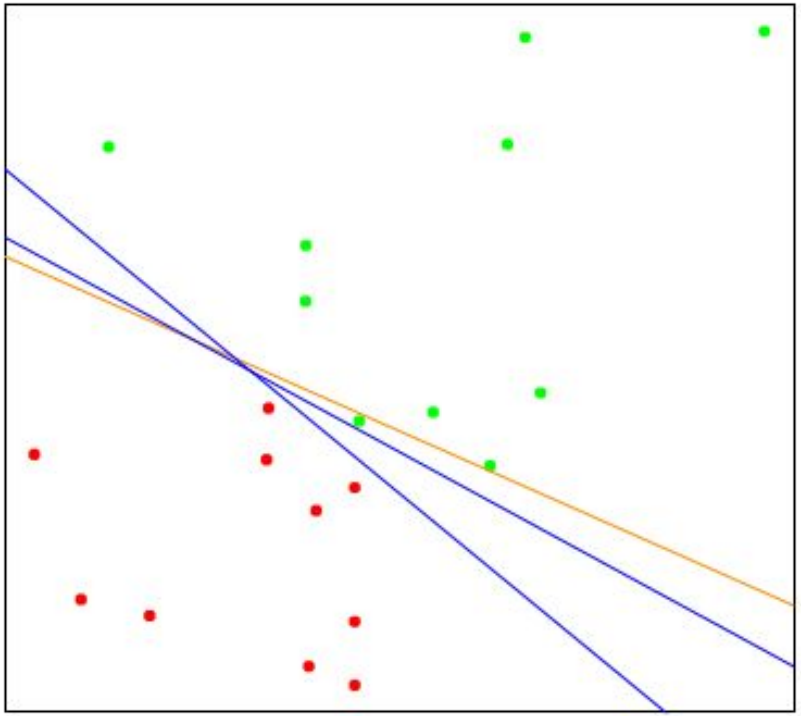
Funciones discriminantes

- Una función discriminante toma un vector input X (sample) con “n” features, y le asigna una de las K clases, llamada Ck.
- Cuando Ck = 2 -> binaria
- Cuando Ck > 2 = multiclase



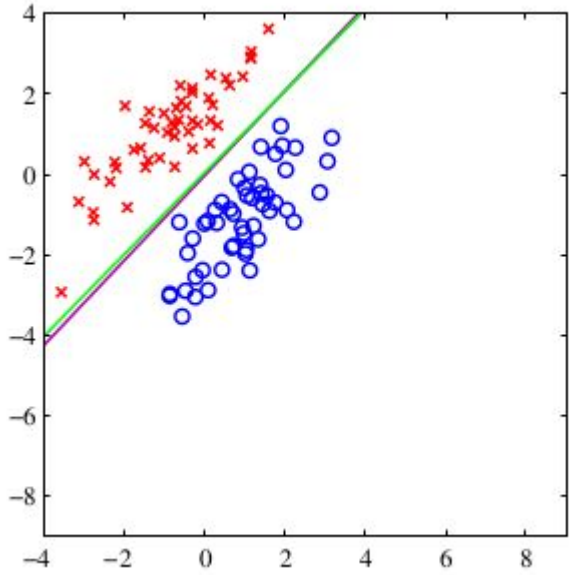
Funciones de Decisión

Para un mismo set de datos etiquetados, distintos modelos pueden generar distintas funciones de decisión (decision rules).

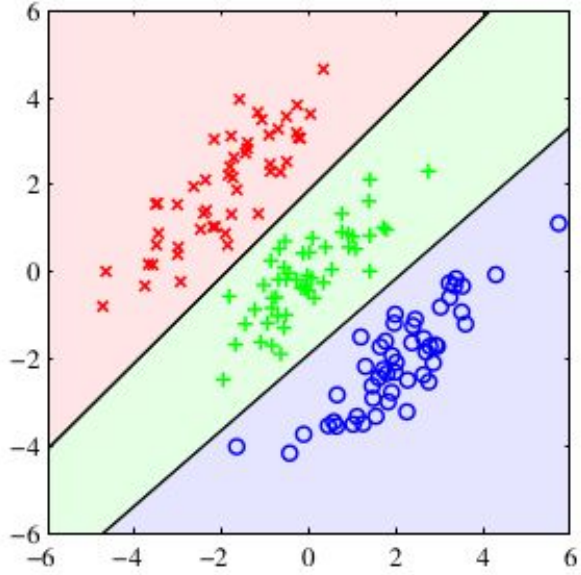


Tipos de clasificación

Clasificación Binaria

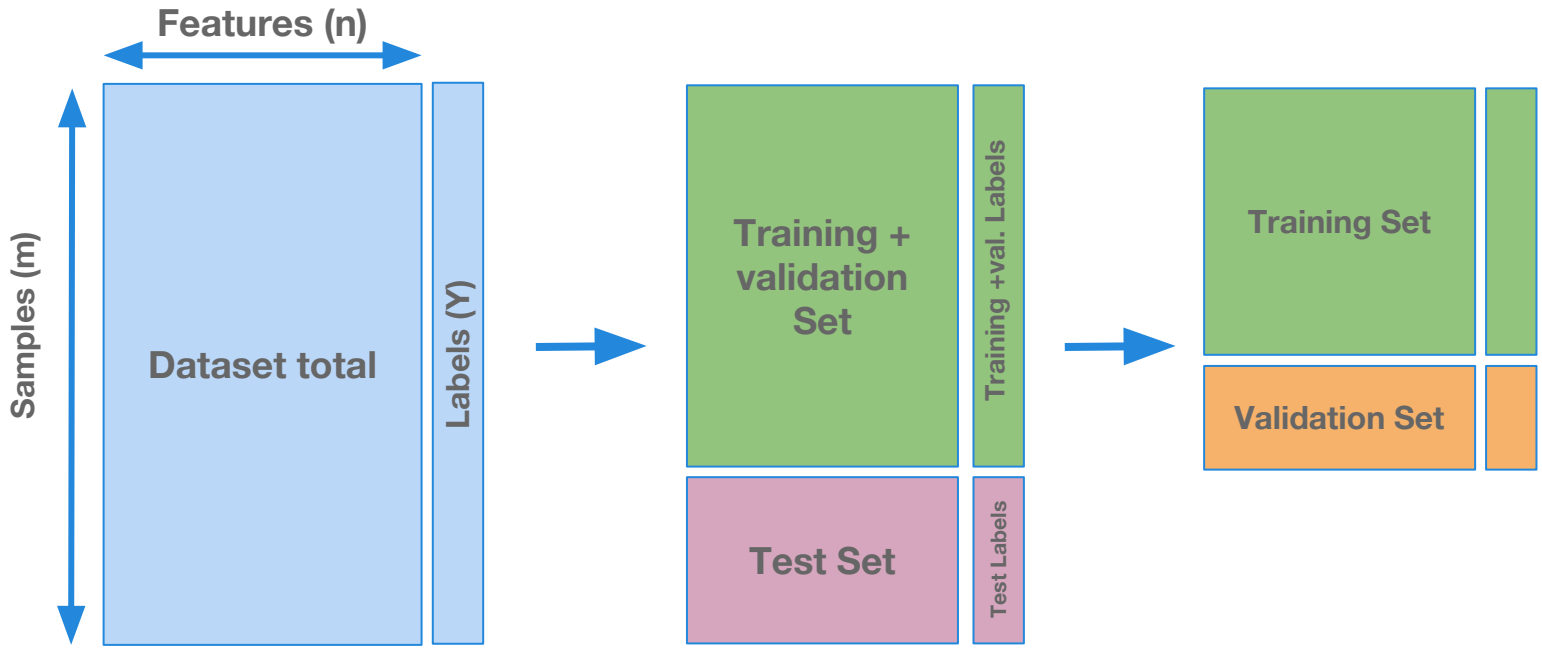


Clasificación Multiclase



*Bishop, Pattern Recognition

Train, Validation, Test sets.

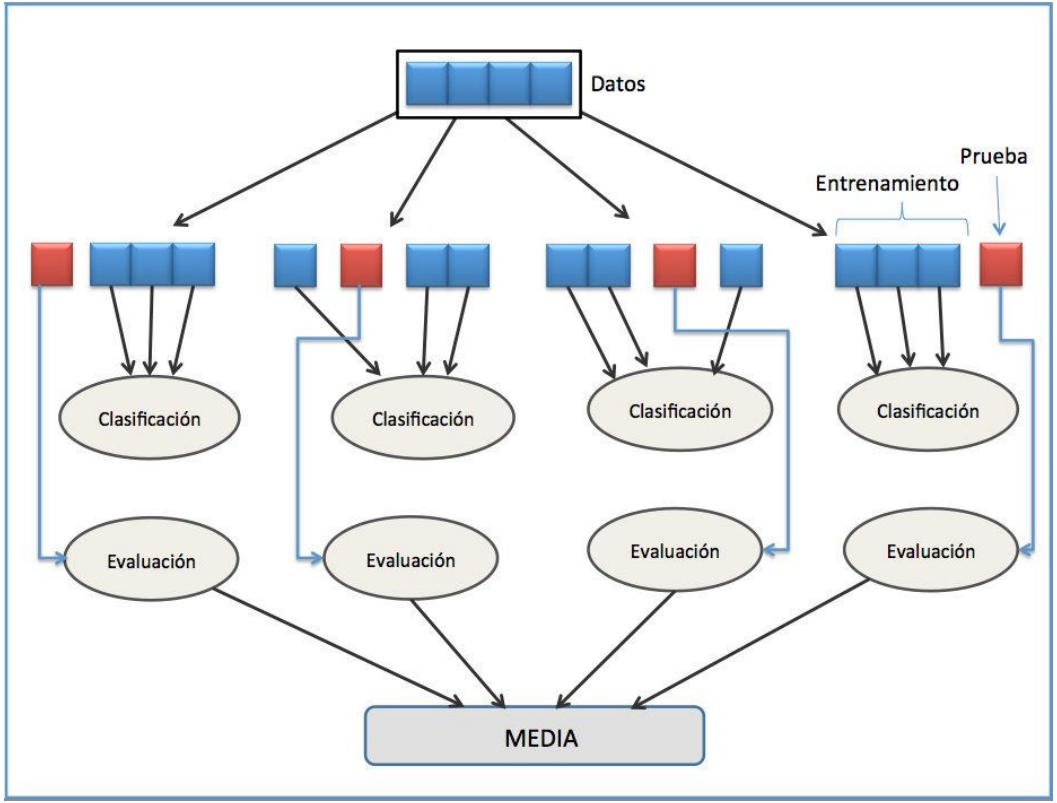


El clasificador aprenderá la regla de decisión utilizando el train set (samples + labels). Luego clasificará las muestras de test (sin mirar las labels de test) y se medirá la exactitud de clasificación en testeo.

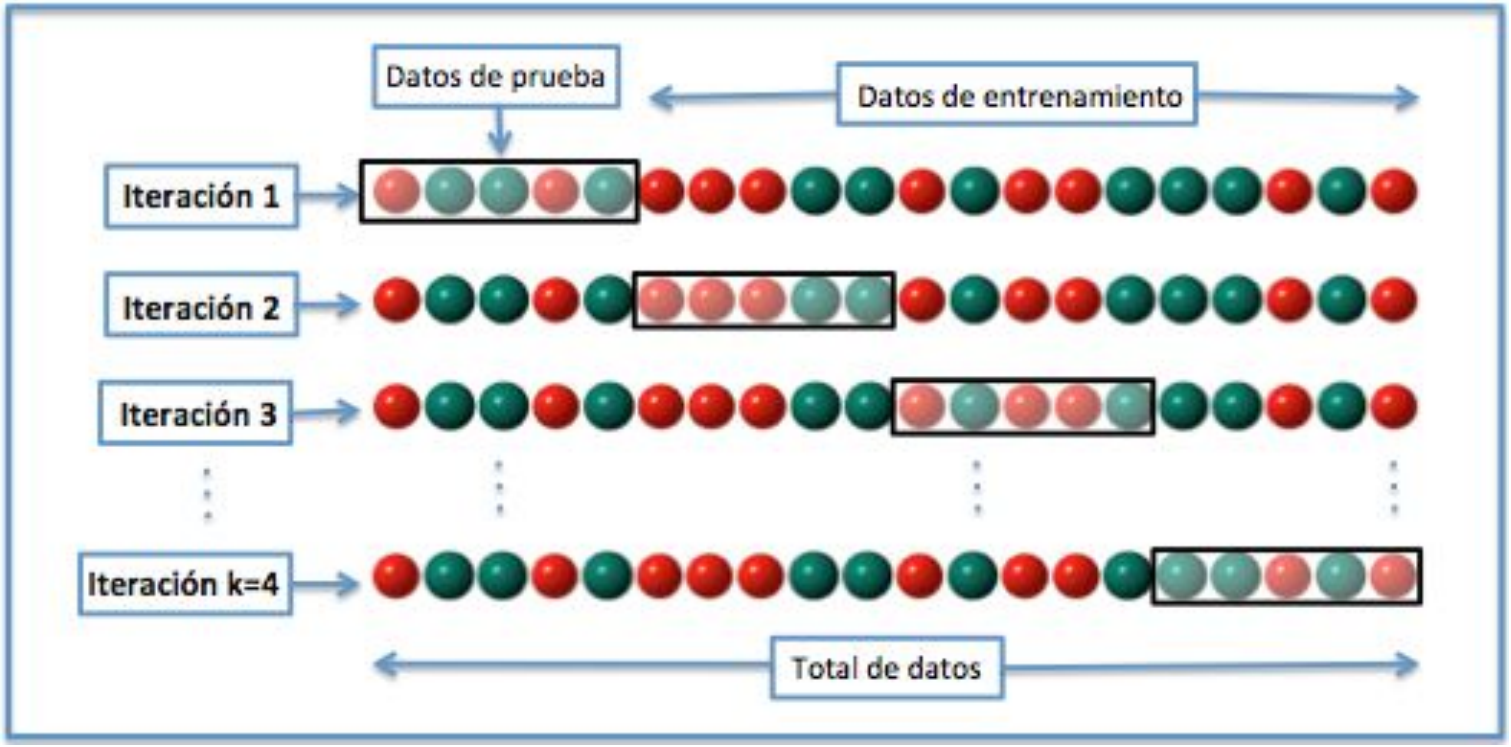
Cross - Validation en training set

Cross validation (CV) se realiza con las muestras de entrenamiento. Consiste en dividir nuestro training set en K folds (K porciones) e iterar K veces. En cada iteración, una porción se utiliza como test y el resto como train. En cada iteración se evaluará el resultado de clasificación y luego se realizará un promedio de todas las iteraciones.

El objetivo es asegurar que nuestro modelo no realiza **overfitting**.

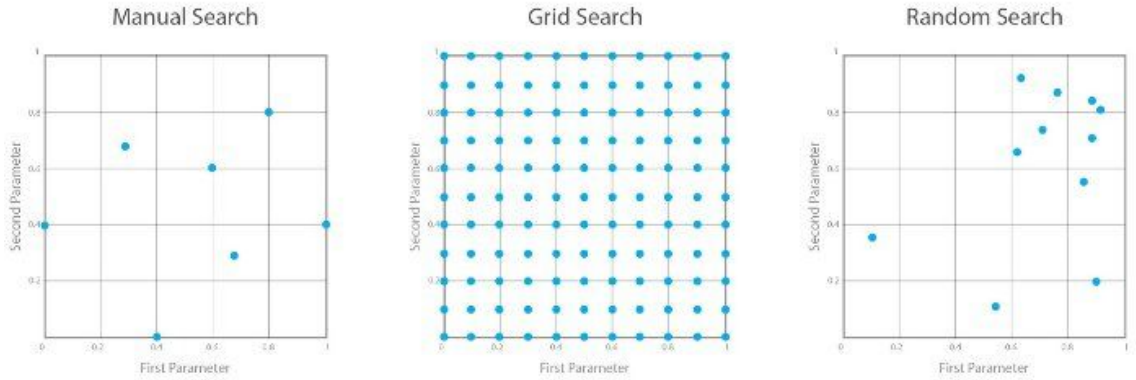


Cross Validation

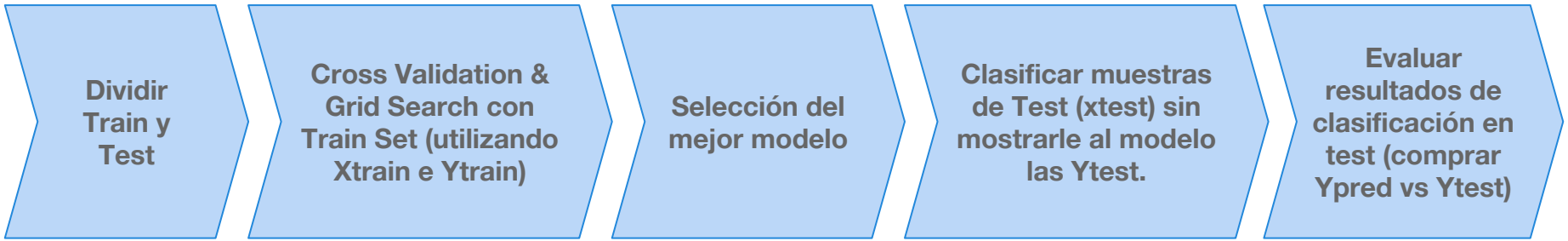


Grid Search

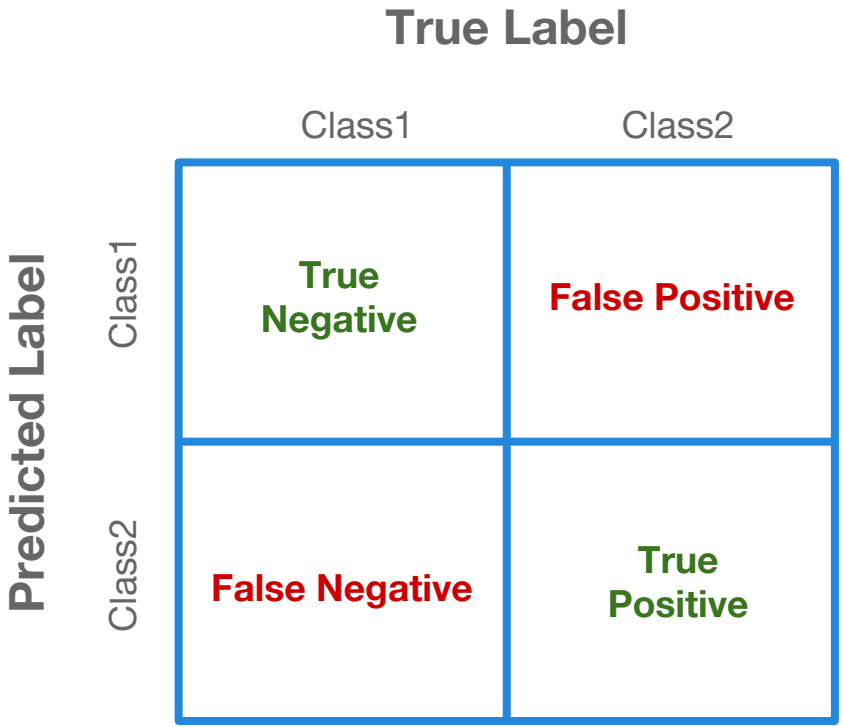
- Los modelos de clasificación que utilizaremos consistirán de **hiperparámetros** que el usuario debe seleccionar. Estos determinarán la regla de decisión y por ende la performance del modelo.
- Para saber qué hiperparámetros seleccionar, lo que haremos es generar una lista de los mismos y probaremos todas las combinaciones posibles de ellos.
- La combinación que mayor Cross-Validation y Train Accuracy genere es la que usaremos para testear el modelo.



Pipeline: Train, Validate, Test Model

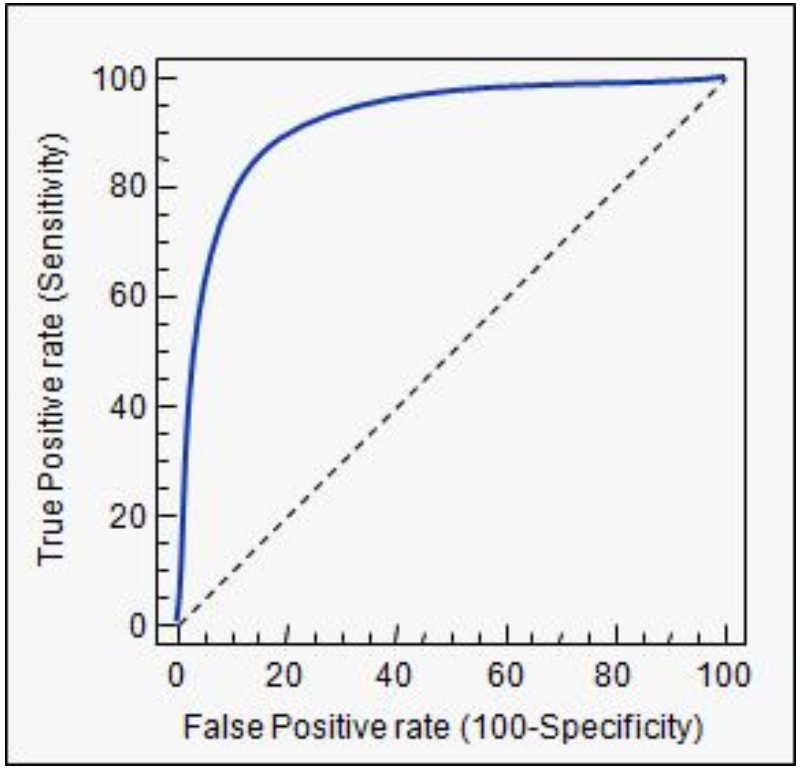


classification results: confusion matrix



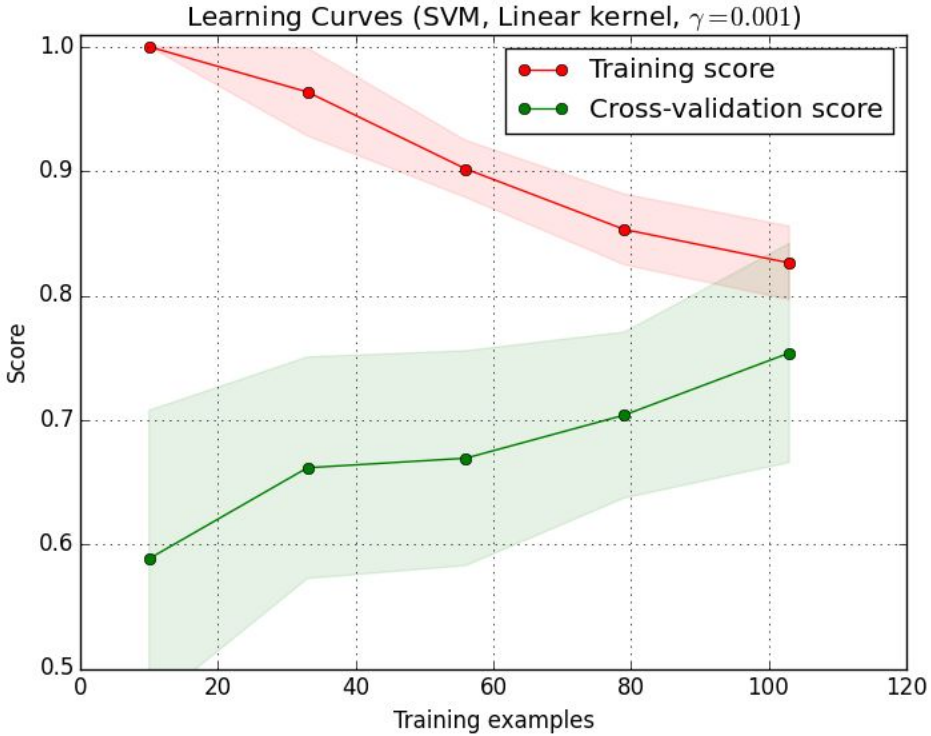
Accuracy = (TN+TP) / Total

classification results: AUC ROC

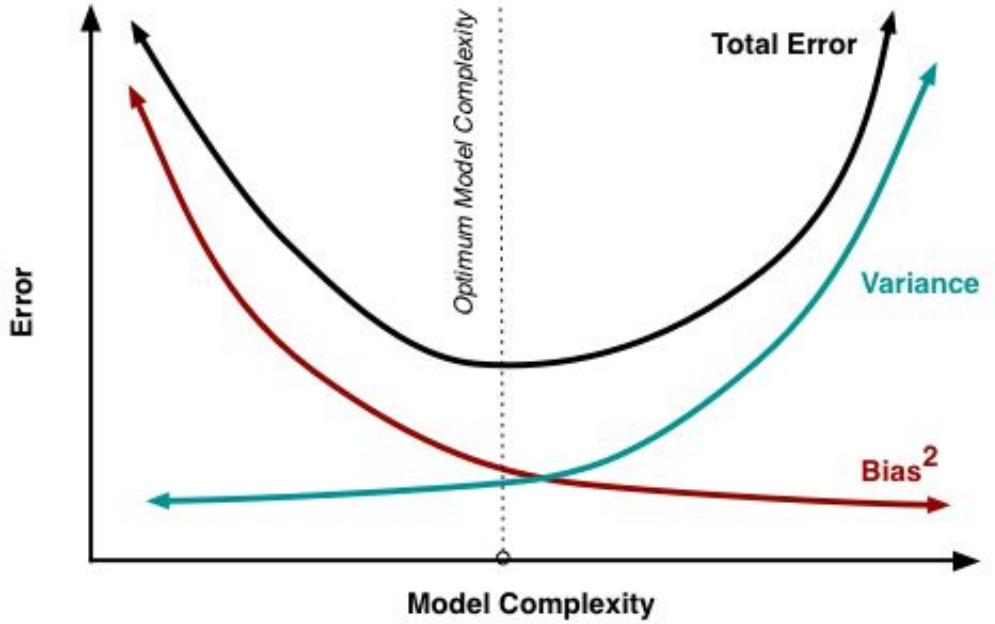


- El área bajo la curva ROC (AUC) da una idea de cuan bueno es mi clasificador independientemente del accuracy.
- Relaciona cuán bien se clasifica ambas clases. Cuanto más cerca de 1 sea el área bajo la curva, mejor mi clasificador.
- Cuanto más cerca de 0.5 sea el AUC, más similar a “arrojar una moneda” sera mi clasificador.

Learning Curve

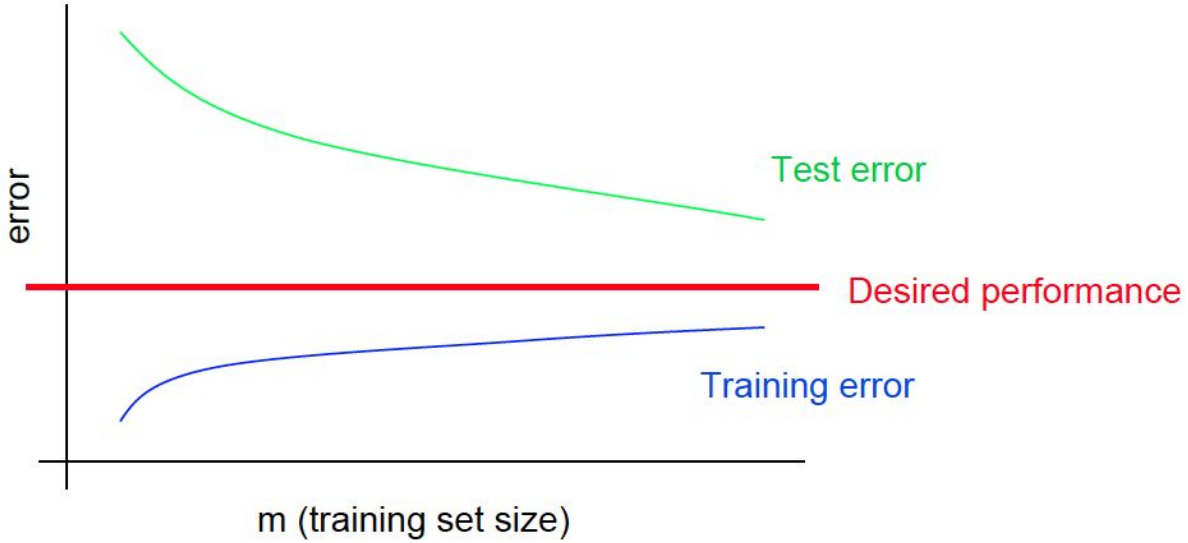


Variance vs Bias



*Andrew Ng.

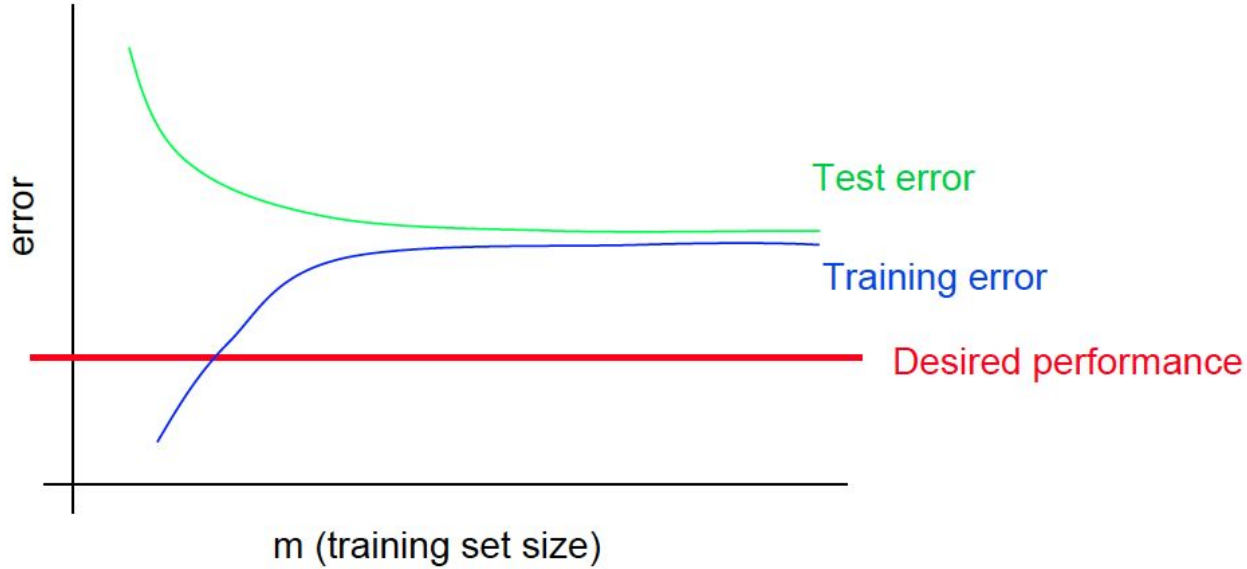
Learning curve: high variance



High variance means lower model complexity (aka, less features)

*Andrew Ng.

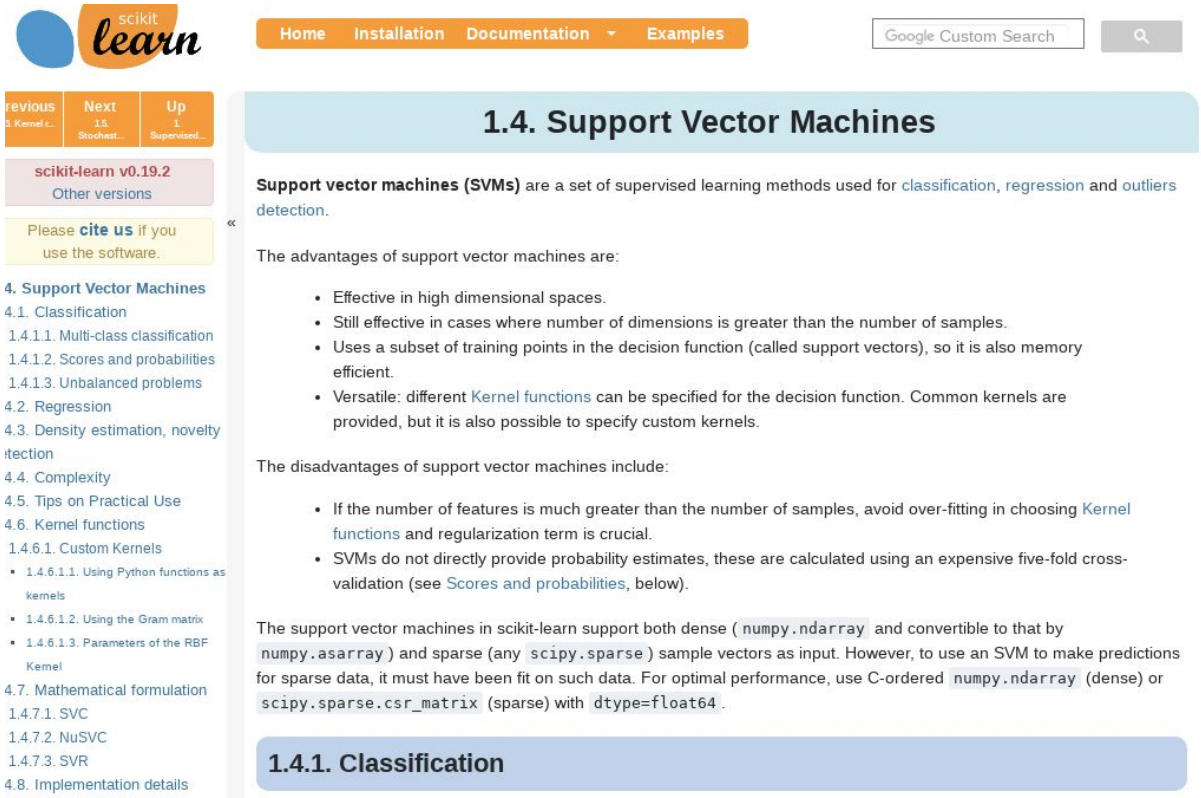
Learning curve: high bias



High bias means higher model complexity (aka, too many features)

*Andrew Ng.

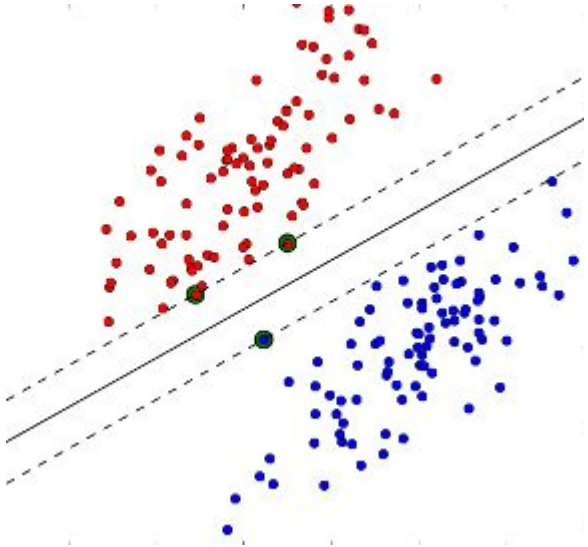
Support Vector Machines



Classification Models: SVM

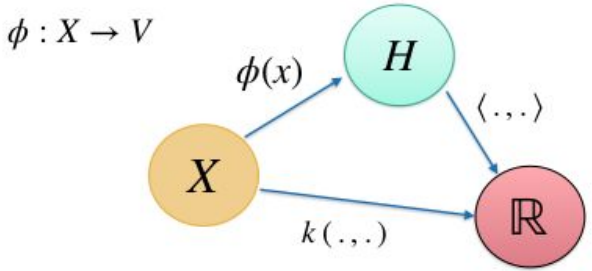
Support Vector Machines

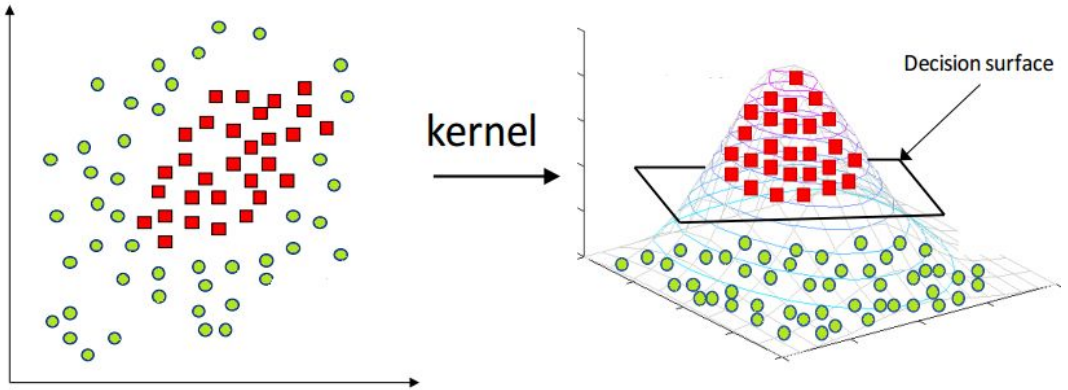
- Clasificador Lineal
- Busca el hiperplano separador maximizando el margen entre clases.
- Cuando las clases no son separables linealmente se acude al “soft-margin”, penalizador que permite muestras “del otro lado”.
- El margen separador queda definido por “s” muestras. Estas muestras son llamadas support vectors.



Classification Models: SVM “Kernel Trick”

$$k\left(x_i, x_j\right) = \left\langle \phi(x_i), \phi(x_j) \right\rangle_H$$





Los kernels son funciones de similaridad entre muestras. Mapean nuestros datos a una dimensión desconocida donde son linealmente separables. Cuando usamos SVM, podemos aplicar un kernel para facilitar la clasificación.

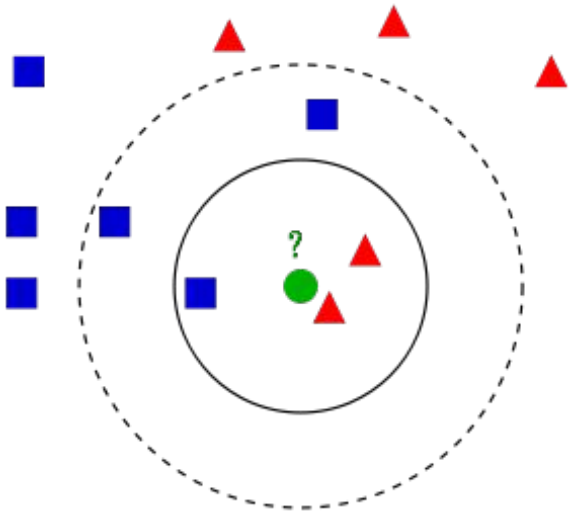
SVM: Parametros

- C = “Costo” (todos los kernels)
- Gamma = Kernel Gaussiano (RBF)
- Degree = Kernel Polynomial


KERNELS
HD
HISTORY.COM

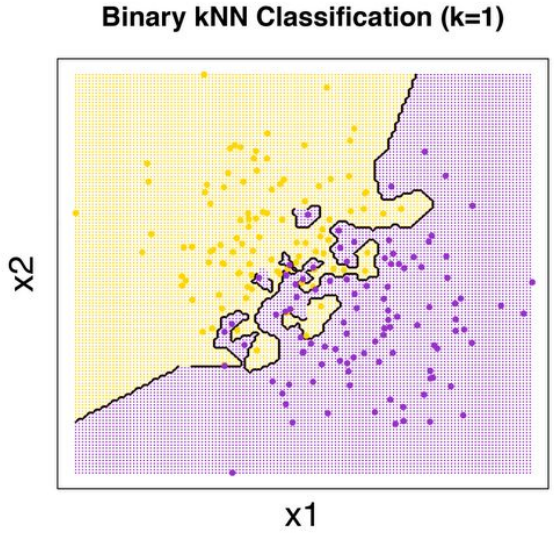
Classification Models: KNN

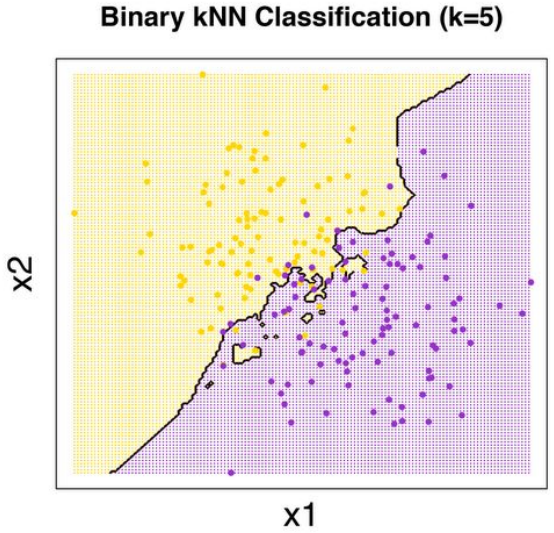
- Clasifica cada nuevo dato en el grupo que corresponda, según tenga **K** vecinos más cerca de un grupo o del otro.
- Calcula la ***distancia*** del elemento nuevo a cada uno de los existentes y ordena esas *distancias* para seleccionar a qué grupo al que pertenece.
- Selecciona la etiqueta (Y) qué más frecuente aparece en las K clases.

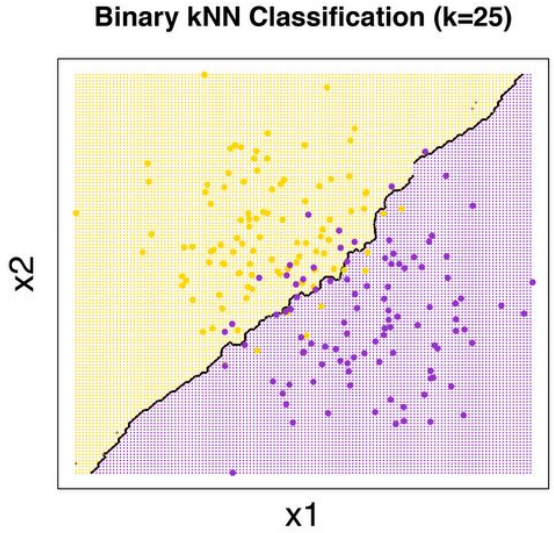


$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

KNN: variar el parametro K







Manos a la obra