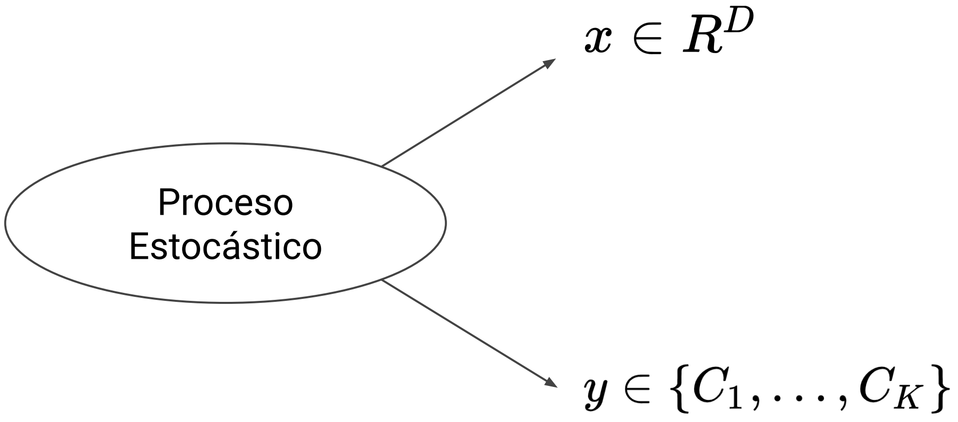
Intro. a Clasificación y Modelos Generativos

I302 - Aprendizaje Automático y Aprendizaje Profundo

Roberto Bunge

Universidad de San Andrés

Problema de Clasificación



- K clases disjuntas
- Dada una muestra x, asignarla a una clase C_j

Ejemplos de Problemas de Clasificación

1. Diagnóstico de una enfermedad
2. Detección de caracteres en una imagen
3. Identificación de personas
4. Diagnóstico de severidad de una enfermedad

Ejemplos de Problemas de Clasificación

1. Diagnóstico de una enfermedad
2. Detección de caracteres en una imagen
3. Identificación de personas
4. Diagnóstico de severidad de una enfermedad $\longrightarrow$ NO! Este es un problema de **regresión con variable de salida discreta, que podemos modelar con una variable real, y luego asignar el valor más cercano (redondear)**

Variables Aleatorias Categóricas

- ¿Cómo podemos representar una variable aleatoria categórica?
- Si K = 2 $\rightarrow$ $y \in \{0, 1\}$ variable binaria
- Si K > 2 $\rightarrow$ $y \in \mathbb{N}^K$ variable multi-clase ("one-hot encoding")

  $0 \leq y_j \leq 1$

  $\sum_{j=1}^{K} y_j = 1$

O sea, vector con todos los elementos cero, excepto uno que está "encendido"

- Esta representación se puede aplicar tanto a las variables de salida, como las de entrada!

Asignación de Clases

- Tenemos que dividir el espacio de entradas x, en regiones disjuntas

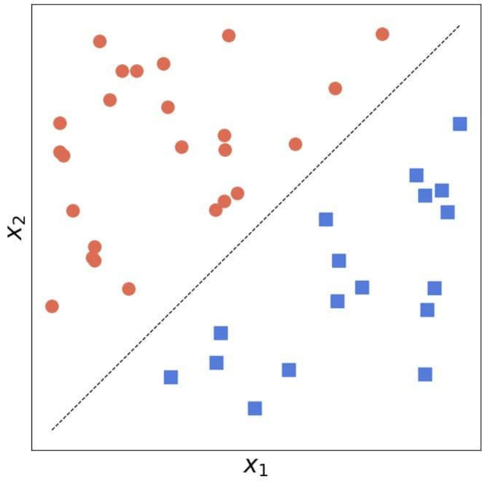
$$R_1, \ldots, R_K$$

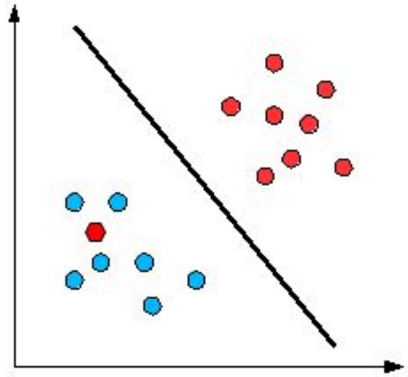
- Si x cae dentro de la region R_j, entonces le asignamos la clase C_j

$$x \in R_j \rightarrow \hat{y} = C_j$$

Separabilidad Lineal

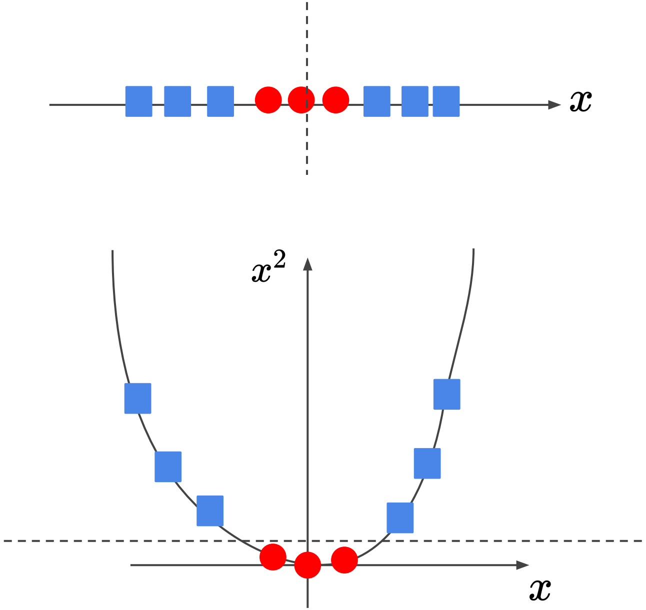
- Existe un hiperplano en el espacio de features tal que todos los puntos de una clase quedan de un lado del hiperplano en espacio de features, y los de otra clase quedan del otro
- Dependiendo del proceso estocástico y los features, un conjunto de datos puede o no ser separable

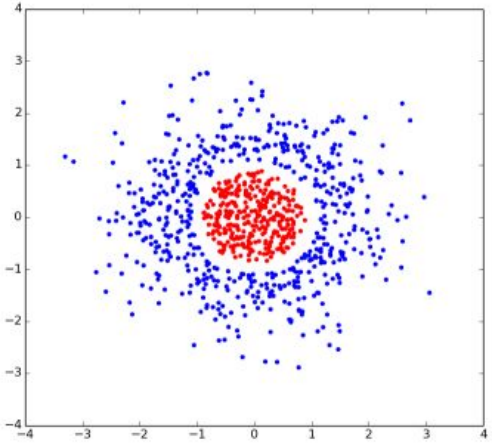


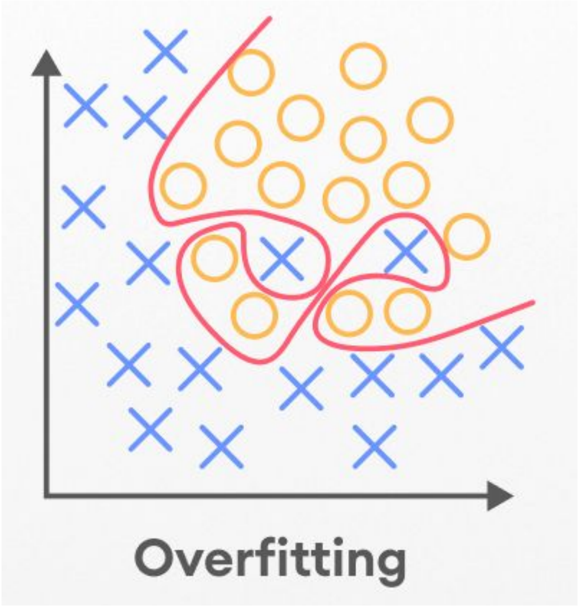


Separabilidad Lineal

- A medida que agregamos features, los datos tienden a ser más separables
- Algunos features hacen que sea mas separable que otros

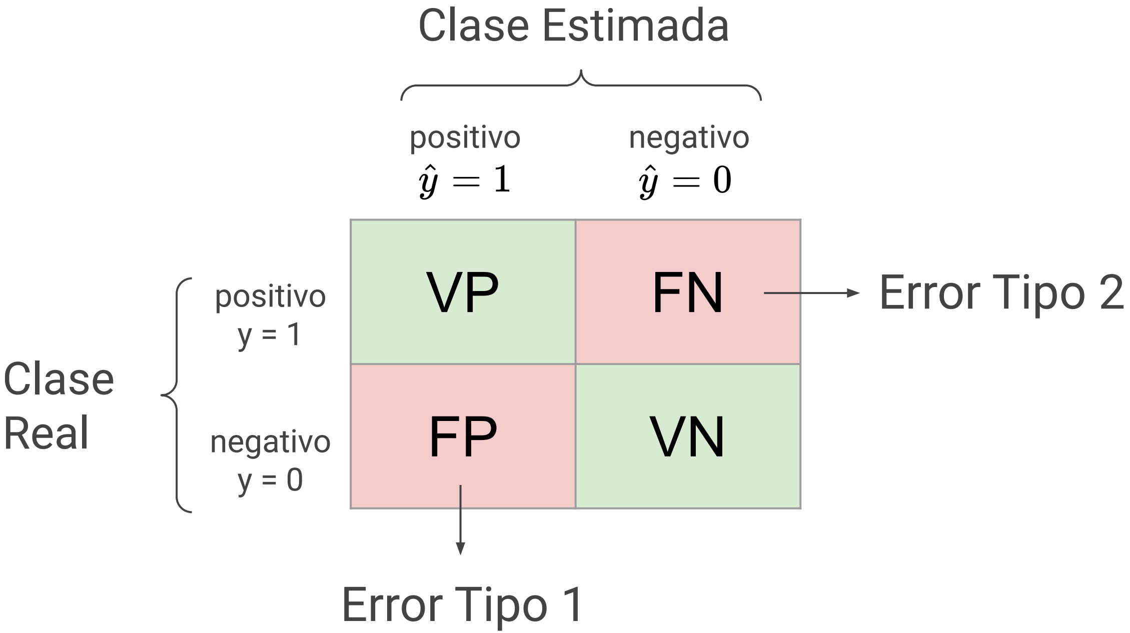






Métricas de Performance (clasificación binaria)

- Matriz de confusión:



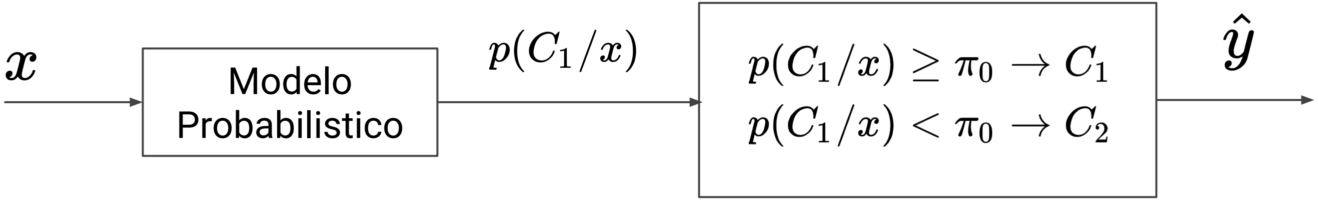
- Idealmente, FN = 0 y FP = 0
  - Si es separable, entonces se puede!
  - Si no es separable, hay un "trade-off" entre FN y FP, cual priorizo?

Métricas de Performance (clasificación binaria)

- Accuracy = $\frac{VP+VN}{VP+VN+FP+FN}$ ¿Sobre el total, cuantos clasificó correctamente?
- Recall (true positive rate) = $\frac{VP}{VP+FN}$ ¿De los positivos reales, cuantos detectó como positivos?
- Precision = $\frac{VP}{VP+FP}$ ¿Cuando clasifica positivo, qué fracción es realmente positivo?
- False positive rate = $\frac{FP}{FP+VN}$ ¿De los negativos reales, cuantos detectó como positivos?
- False discovery rate = $\frac{FP}{FP+VP}$ ¿Cuando clasifica positivo, qué fracción es realmente negativo?

Umbral de Clasificación

- Un clasificador probabilístico genera una probabilidad posterior
- Esto es convertido a una asignación de clase mediante un umbral

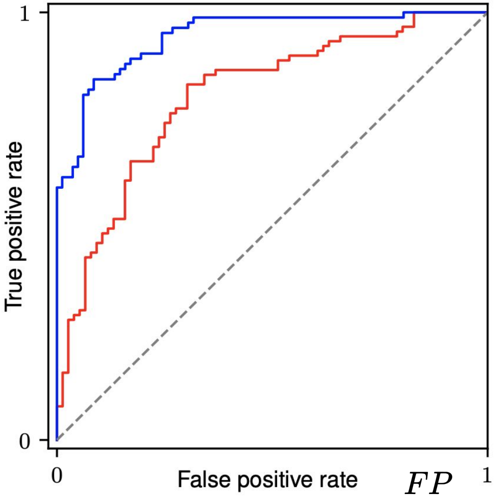


- Al variar el umbral podemos reducir los errores de tipo 1 a expensas de incrementar errores tipo 2

Curva ROC

- Podemos graficar como variar el "true positive rate" vs. "false positive rate", a medida que cambiamos el umbral de 0 a 1

$$\frac{VP}{VP+FN}$$

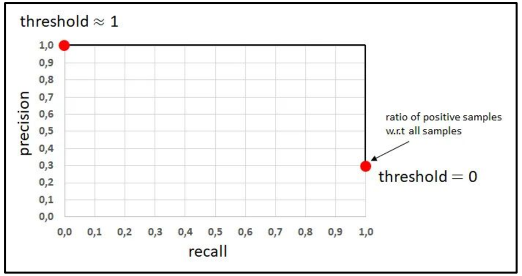


$$\frac{FP}{FP+VN}$$

- Esto se conoce como la curva ROC (Receiver operating characteristics)

Curva PR

- Podemos graficar como variar el "precision" vs. "recall", a medida que cambiamos el umbral de 0 a 1



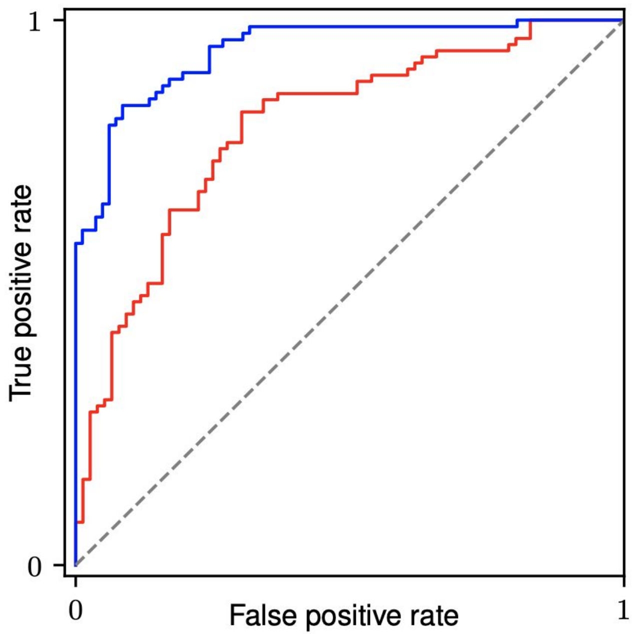
Clasificador perfecto

Clasificador aleatorio

- Esto se conoce como la curva PR

ROC y "área bajo la curva"

- Se puede reducir una curva ROC a un número, tomando la integral (area) bajo la curva
- AUC = area under the curve
- AUC más grande es en general mejor
- Clasificador perfecto
    - AUC-ROC = 1 = AUC-PR
- Clasificador random:
    - AUC-ROC = 0.5
    - AUC-PR = proporcion de positivos sobre total



F-Score

- Combina precision y recall para dar un solo número

- F-score es el promedio geométrico de precision y recall

$$F = \frac{2 \times precision \times recall}{precision + recall}$$
$$= \frac{2VP}{2VP + FP + FN}$$

Loss function

- Otra opción es asignar un costo a cada tipo de error y armar una función de costo total
- Matriz de costo:

0	λ_{FN}
λ_{FP}	0

- Función de costo total:

$$L = \lambda_{FP} FP + \lambda_{FN} FN$$

- Al variar el costo relativo, vamos a hacer el trade-off entre FP y FN
  - Podemos trazar una curva pareto variando los costos relativos!

Tipos de Modelos de Clasificación

Función discriminante, funcion que asigne directamente x a clase

Modelos generativos, modelar $\begin{cases} p(x/C_j) \\ p(C_j) \end{cases}$ $\Rightarrow$ $p(C_j/x) = \frac{p(x/C_j)p(C_j)}{\sum_{j=1}^{K} p(x/C_j)p(C_j)}$

Modelos discriminativos, modelar directamente $p(C_j/x)$

Gaussian Discriminant Analysis (GDA)

1. Asumir que las entradas condicionadas por clase siguen una distribución Gaussiana multivariable :

$$p(x/C_j) = \mathcal{N}(\mu_j, \Sigma_j)$$

2. Cada clase tiene un cierto prior:

$$p(C_j) = \pi_j$$

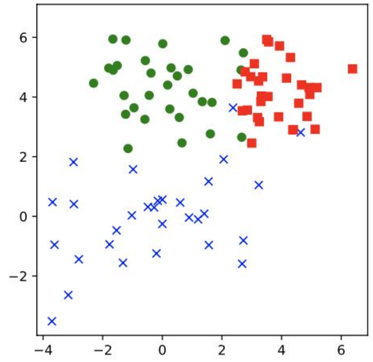
Gaussian Discriminant Analysis (GDA)

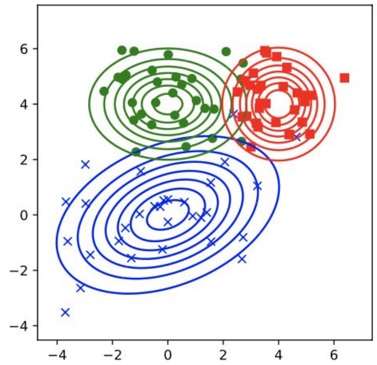
- Aplicando el principio de máxima verosimilitud sobre un set de datos de entrenamiento, podemos ajustar los parámetros del modelo

$$\pi_j^* = \frac{N_j}{N}$$

$$\mu_j^* = \frac{1}{N_j} \sum_{i \in C_j} x_i$$

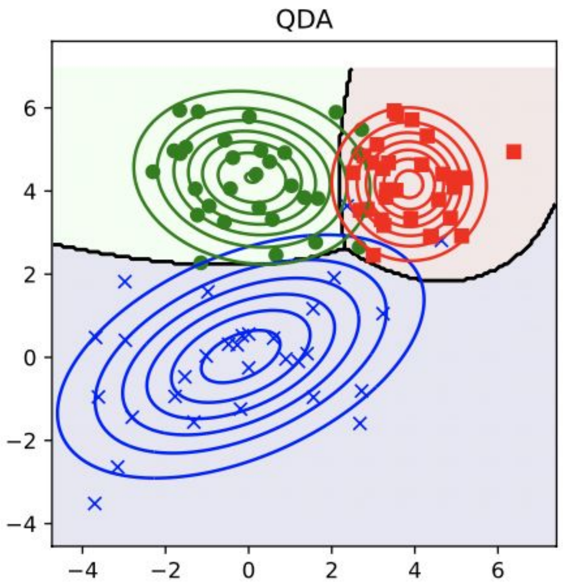
$$\Sigma_j^* = \frac{1}{N_j} \sum_{i \in C_j} (x_i - \mu_j^*)(x_i - \mu_j^*)^T$$





Gaussian Discriminant Analysis (GDA)

- Dado un nuevo x, podemos computar la probabilidad de las clases a posteriori, y asignar la clase más probable
- Si observamos la forma de los log-likelihoods, vemos que la frontera de decisión es cuadrática



$$\ln(p(C_j/x,w)) = \ln\pi_j - \frac{1}{2}\ln|2\pi\Sigma_j^*| + 0.5(x-\mu_j^*)^T {\Sigma_j^*}^{-1}(x-\mu_j^*) + cte$$

Linear Discrminant Analysis (LDA)

- Si no tenemos suficientes datos para aprender los parámetros de GDA, podemos reducir los parámetros asumiendo que las matrices de covarianza son iguales para cada clase

$$\Sigma_j = \Sigma$$

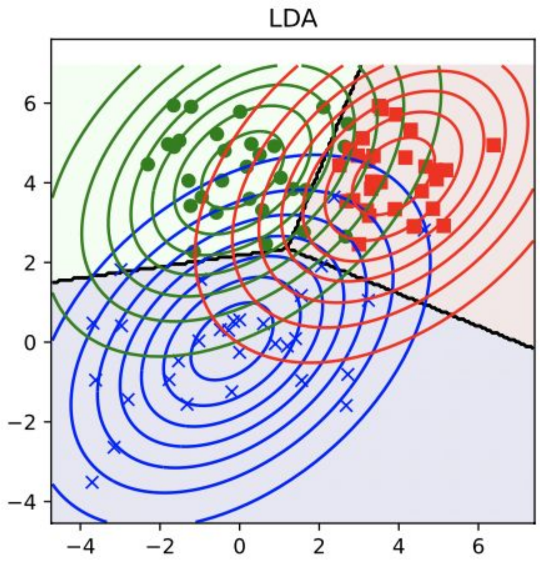
- Aplicando maxima verosimilitud:

$$\Sigma^* = \frac{1}{N} \sum_{j=1}^{K} N_j \Sigma_j^*$$

Linear Discriminant Analysis (LDA)

- Dado un nuevo x, podemos computar la probabilidad de las clases a posteriori, y asignar la clase más probable

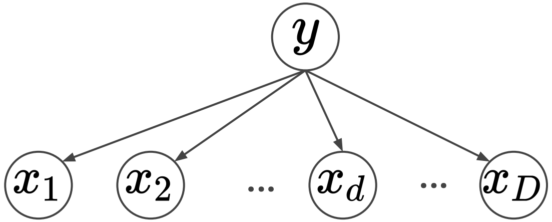
- Si observamos la forma de los log-likelihoods, vemos que la frontera de decisión es lineal (porque las covarianzas de la Gaussiana son iguales)



$$\ln(p(C_j/x,w)) = \ln \pi_j - \frac{1}{2}\ln|2\pi\Sigma^*| + 0.5(x-\mu_j^*)^T {\Sigma^*}^{-1}(x-\mu_j^*) + cte$$

Naive Bayes

- Se puede simplificar significativamente la complejidad del modelo si asumimos que las features son condicionalmente independientes, dada la clase ("Naive Bayes Assumption")



$$p(x/C_j, w) = \Pi_{d=1}^{D} p(x_d/C_j, w_{dj})$$

Naive Bayes

- Si una **feature x_d es binaria**, la puedo modelar con distribución Bernoulli:

$$p(x_d = 1/C_j) = w_{dj}$$

- Aplicando máxima verosimilitud sobre el set de entrenamiento:

$$w^*_{dj} = \frac{N_{dj}}{N_j}$$

la cantidad de veces que x_d estuvo encendida (x_d = 1), dentro de las muestras pertenecientes a la clase j

la cantidad de muestras pertenecientes a la clase j

Naive Bayes

- Si una **feature x_d es real**, se puede modelar con una distribución Gaussiana:

$$p(x_d = 1/C_j) = \mathcal{N}(\mu_{dj}, \sigma^2_{dj})$$

- Aplicando máxima verosimilitud sobre el set de entrenamiento:

$$\mu^*_{dj} =$$ media muestral sobre los x_d pertenecientes a clase C_j

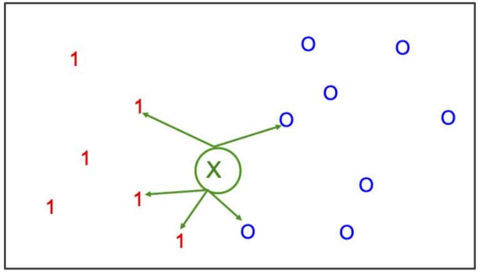
$${\sigma^2}^*_{dj} =$$ varianza muestral sobre los x_d pertenecientes a clase C_j

K-nearest neighbours

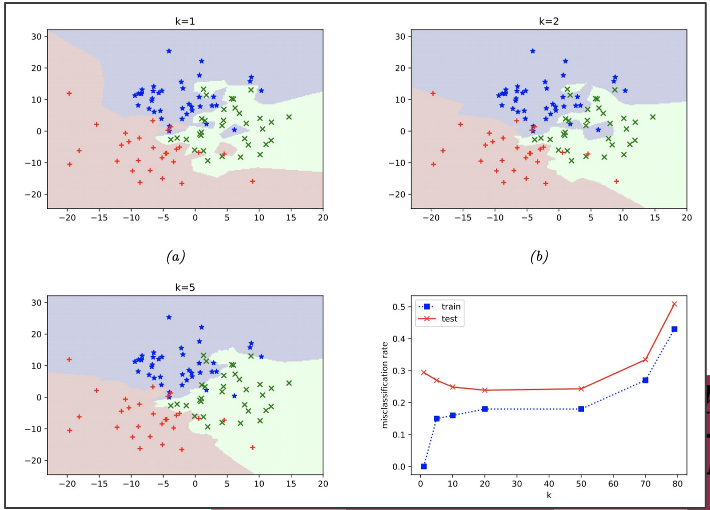
$$p(x/C_j) = \frac{K_j}{N_j V}$$

$$p(x) = \frac{K}{NV}$$

$$p(C_j) = \frac{N_j}{N}$$



- KNN puede ser interpretado como un clasificador generativo
- Parámetro del modelo: K
    - No confundir con la cantidad de clases K!
- V: volumen de la esfera centrada en x, que contiene K muestras
- K_j: de las K muestras, la cantidad que pertenece a la clase j
- N: cantidad de muestras totales
- N_j: de las muestras totales, las que pertenecen a la clase j



$$\frac{K_j}{K}$$

K-nearest neighbours

- Parámetro del modelo: K
    - No confundir con la cantidad de clases K!
- V: volumen de la esfera centrada en x, que contiene K muestras
- K_j: de las K muestras, la cantidad que pertenece a la clase j
- N: cantidad de muestras totales
- N_j: de las muestras totales, las que pertenecen a la clase j

$$p(x/C_j) = \frac{K_j}{N_j V}$$

$$p(x) = \frac{K}{NV}$$

$$p(C_j) = \frac{N_j}{N}$$

$$p(C_j/x) = \frac{p(x/C_j)p(C_j)}{p(x)} = \frac{K_j}{K}$$