Clasificación

Estadística (M)

El problema de clasificación

Consideremos una variable categórica Y que toma valores 0 y 1, que puede indicar la pertenencia a una categoría o a una clase o a un estado, por ejemplo sano o enfermo y se quiere predecir el estado en función de otras variables $(X_1, ..., X_p)$ por ejemplo: peso, edad, nivel de colesterol, nivel de glucosa en sangre, presión sanguínea.

- Spam o no Spam: queremos clasificar un correo como spam o no de acuerdo a un conjunto de características del correo: presencia de ciertas palabras, país de origen, etc.
- Pago de trajeta de crédito: un banco quiere predecir si un cliente incurrirá en un impago de su tarjeta en base a algunas variables como edad, salario, impagos en los últimos 3 meses, etc.
- Clasificación de hongos: queremos clasificar dos especies de hongos de acuerod a sus características morfológicas.

Clasificador

- Información disponible $X = (X_1, X_2, \ldots, X_p) \in \mathcal{X}$.
- Posibles *etiquetas*. Caso binario $\mathcal{Y} = \{0, 1\}$
- Posibles *etiquetas*. Caso general $\mathcal{Y} = \{y_1, \ldots, y_k\}$
- Clasificador: Regla que asigna a $x \in \mathcal{X}$ un posible valor $y \in \mathcal{Y}$.

Clasificación: Marco Teórico

- $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.
- (X, Y) vector aleatorio, con conjunta F_{XY}.
- Clasificador: Regla (de clasificación) que asigna a cada $x \in \mathcal{X}$ un elemento $y \in \mathcal{Y}$

$$\text{Clasificador} \quad H : \mathcal{X} \to \mathcal{Y}$$

- Error de Clasificación Medio (verdadero - poblacional) del clasificador H

$$L(H) = \mathbb{P}(H(X) \neq Y)$$

- Objetivo (teórico): Encontrar H que minimice el error medio de clasificación.

H^{opt} Optimo: Regla de Bayes - Caso binario

$$H^{opt}(x) = \begin{cases} 1 & \text{si} \quad \mathbb{P}(Y=1 \mid X=x) > \mathbb{P}(Y=0 \mid X=x), \\ 0 & \text{si} \quad \mathbb{P}(Y=0 \mid X=x) > \mathbb{P}(Y=1 \mid X=x). \end{cases}$$

H^{opt} Optimo: Regla de Bayes - Caso binario

$$H^{opt}(x) = \begin{cases} 1 & \text{si} \quad \mathbb{P}(Y=1 \mid X=x) > \mathbb{P}(Y=0 \mid X=x), \\ 0 & \text{si} \quad \mathbb{P}(Y=0 \mid X=x) > \mathbb{P}(Y=1 \mid X=x). \end{cases}$$

Teorema: Para todo H

$$L(H^{opt}) = P\left(H^{opt}(X) \neq Y\right) \leq P\left(H(X) \neq Y\right) = L(H)\,.$$

Error de Clasificación Empírico de H

- $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.
- (X, Y) vector aleatorio, con conjunta F_{XY}.
- Clasificador: Regla (de clasificación) que asigna a cada $x \in \mathcal{X}$ un elemento $y \in \mathcal{Y}$

$$\text{Clasificador} \quad H : \mathcal{X} \to \mathcal{Y}$$

- Error de Clasificación Medio (verdadero - poblacional) del clasificador H

$$L(H) = \mathbb{P}(H(X) \neq Y)$$

Error de Clasificación Empírico de H

- $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.
- (X, Y) vector aleatorio, con conjunta F_{XY}.
- Clasificador: Regla (de clasificación) que asigna a cada $x \in \mathcal{X}$ un elemento $y \in \mathcal{Y}$

$$\text{Clasificador} \quad H : \mathcal{X} \to \mathcal{Y}$$

- Error de Clasificación Medio (verdadero - poblacional) del clasificador H

$$L(H) = \mathbb{P}(H(X) \neq Y)$$

- Datos: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Error de Clasificación Empírico del clasificador H: proporción de pares mal clasificados según H.

$$\widehat{L}_n(H) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{H(x_i) \neq y_i}$$

Error de Clasificación Empírico de $\hat{H}_n$

- $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.
- Datos: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- $\hat{H}_n$: Procedimiento contruido con los datos.

$$\hat{H}_n : \mathcal{X} \to \mathcal{Y}$$

- Error de Clasificación Empírico del clasificador $\hat{H}_n$:

$$\widehat{L}_n(\hat{H}_n) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\hat{H}_n(x_i) \neq y_i}$$

Error de Clasificación Empírico de $\hat{H}_n$

- $X \in \mathcal{X}$, $Y \in \mathcal{Y}$.
- Datos: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- $\hat{H}_n$: Procedimiento contruido con los datos.

$$\hat{H}_n : \mathcal{X} \to \mathcal{Y}$$

- Error de Clasificación Empírico del clasificador $\hat{H}_n$:

$$\widehat{L}_n(\hat{H}_n) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\hat{H}_n(x_i) \neq y_i}$$

Cuidado: Overfitting!

- Error de Clasificación leave one out (cross – validation) de la regla:

$$\text{CV} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\hat{H}_n^{(-i)}(x_i) \neq y_i}.$$

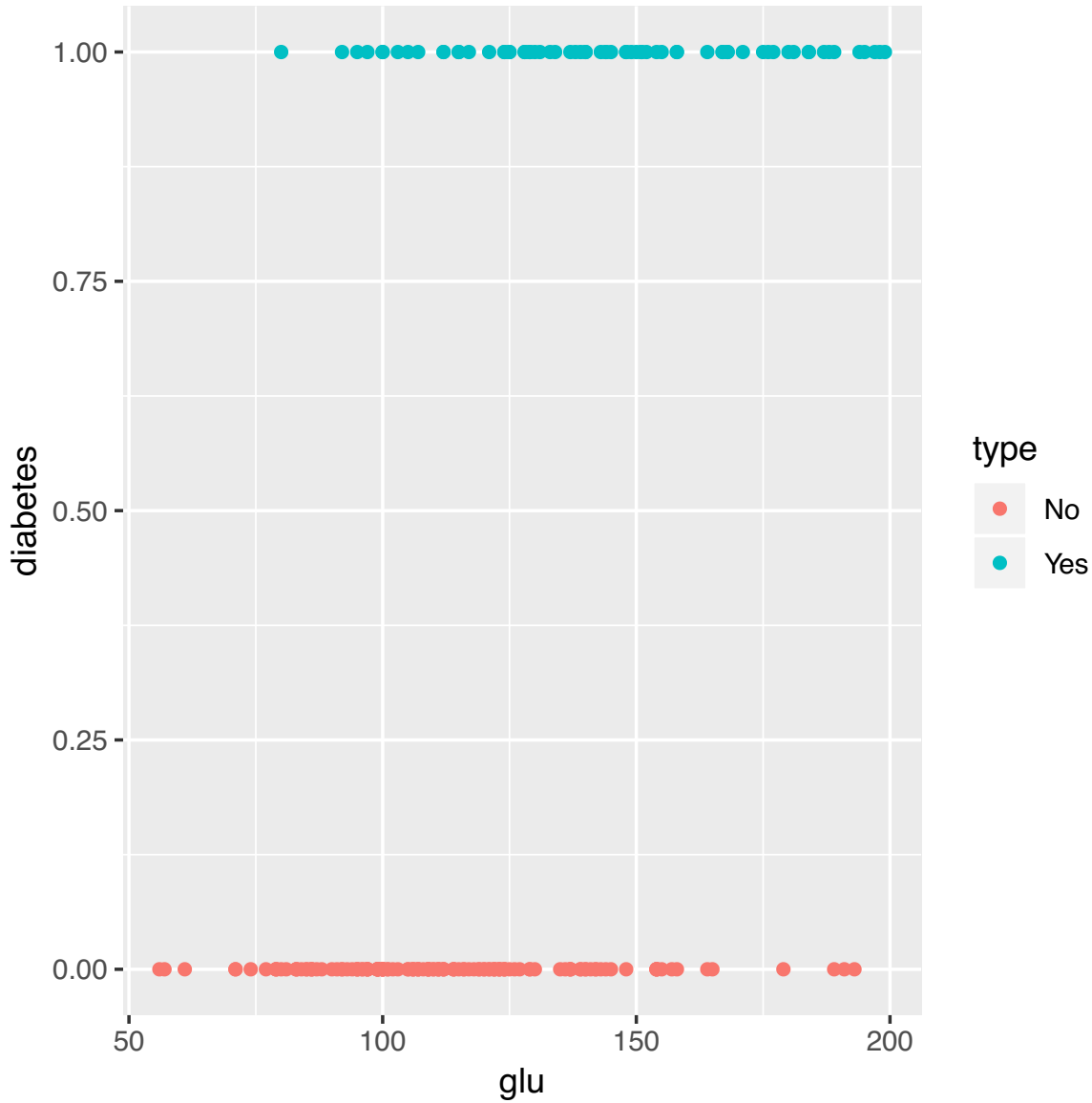
Vayamos un poco para atrás....

- x v.a. $\in \mathcal{X}$
- Posibles *etiquetas*. Caso binario $\mathcal{Y} = \{0, 1\}$
- Clasificador: Regla que asigna a cada $x \in \mathcal{X}$ un elemento $y \in \mathcal{Y}$
- H^{opt} Optimo: Regla de Bayes - Caso binario

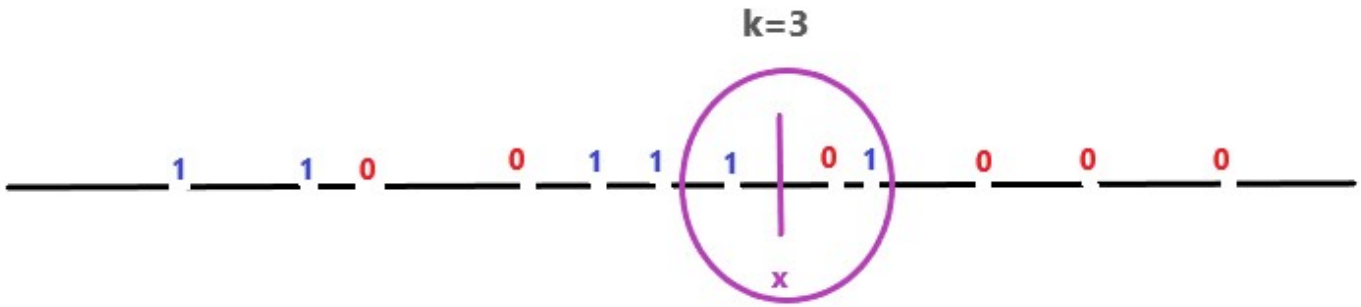
$$H^{opt}(x) = \begin{cases} 1 & \text{si} \quad \mathbb{P}(Y = 1 \mid X = x) > \mathbb{P}(Y = 0 \mid X = x), \\ 0 & \text{si} \quad \mathbb{P}(Y = 0 \mid X = x) > \mathbb{P}(Y = 1 \mid X = x). \end{cases}$$

¿Cómo podríamos estimar $\mathbb{P}(Y = 1 \mid X = x)$ y $\mathbb{P}(Y = 0 \mid X = x)$?

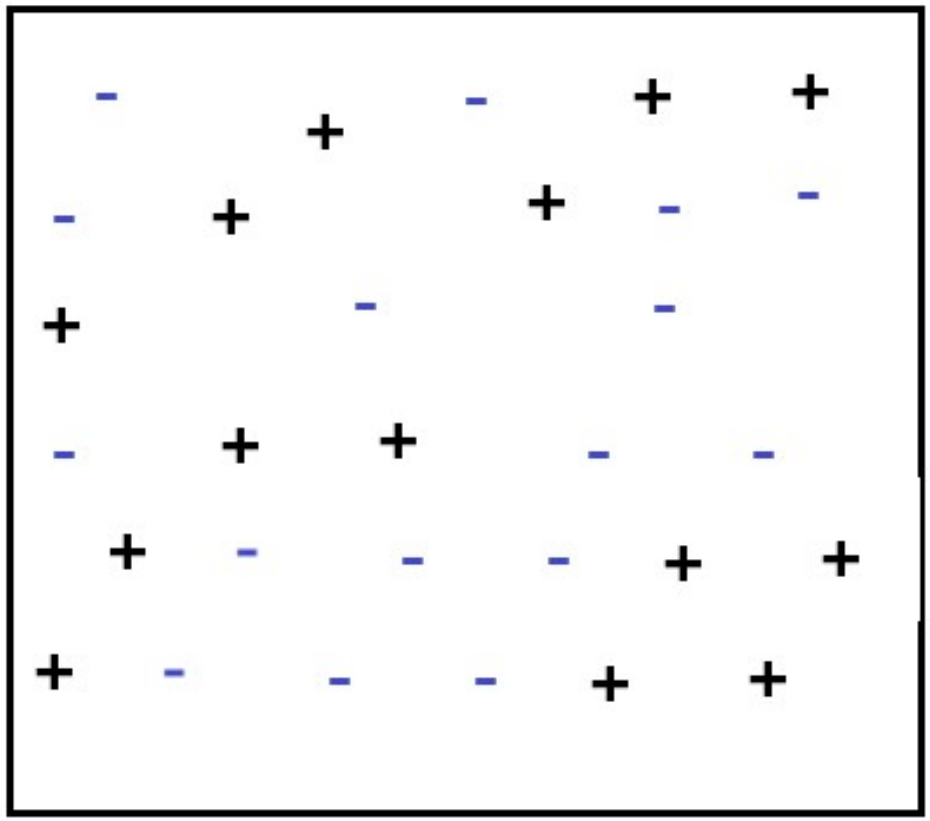
datos PIMA



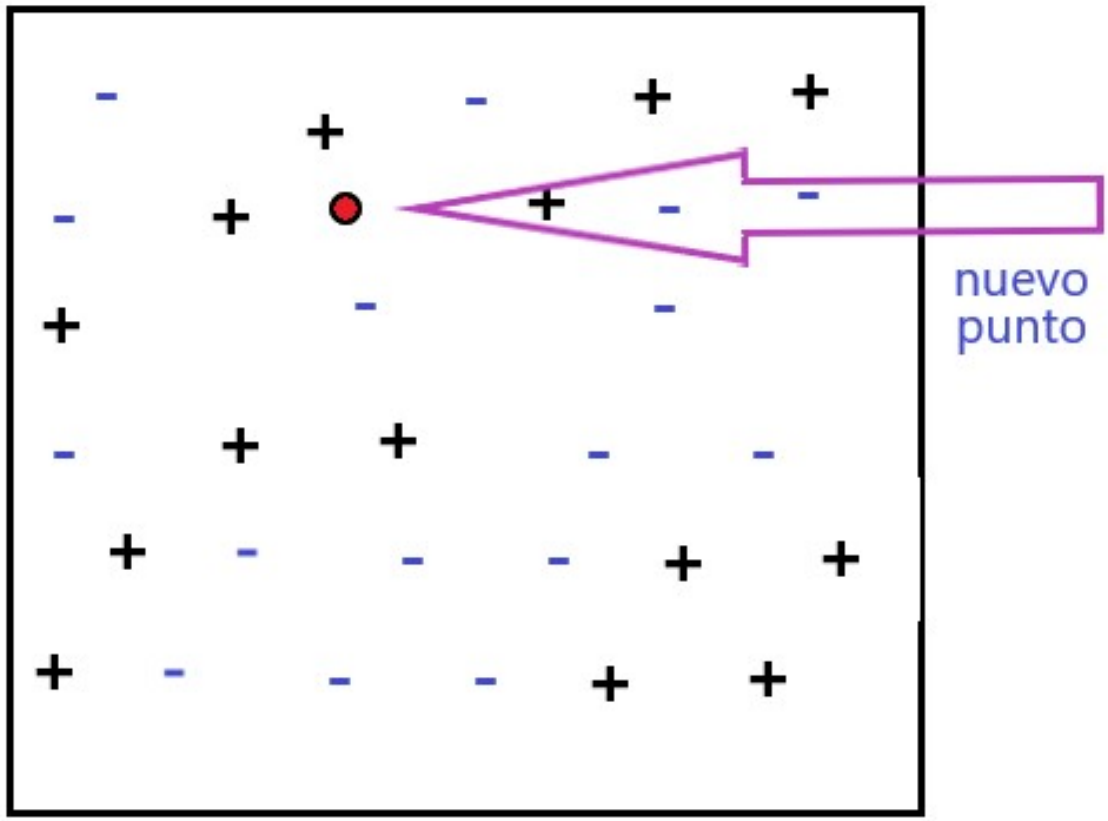
Idea.... :)



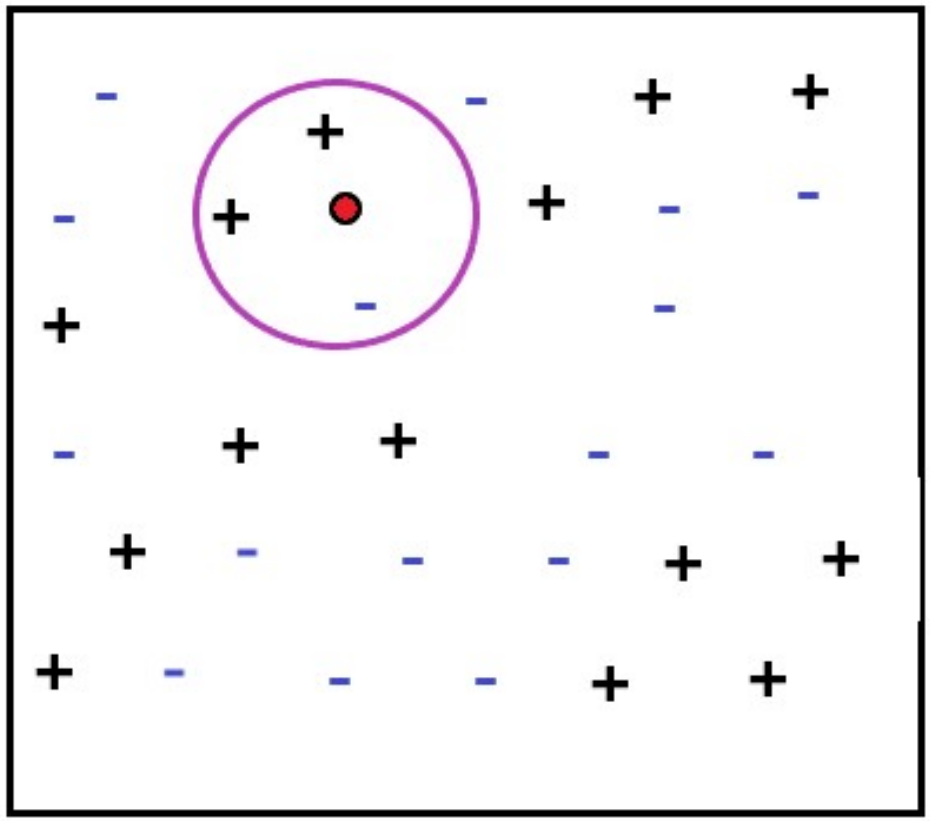
Miremos a los Vecinos más cercanos (kNN: k-nearest neighbors)



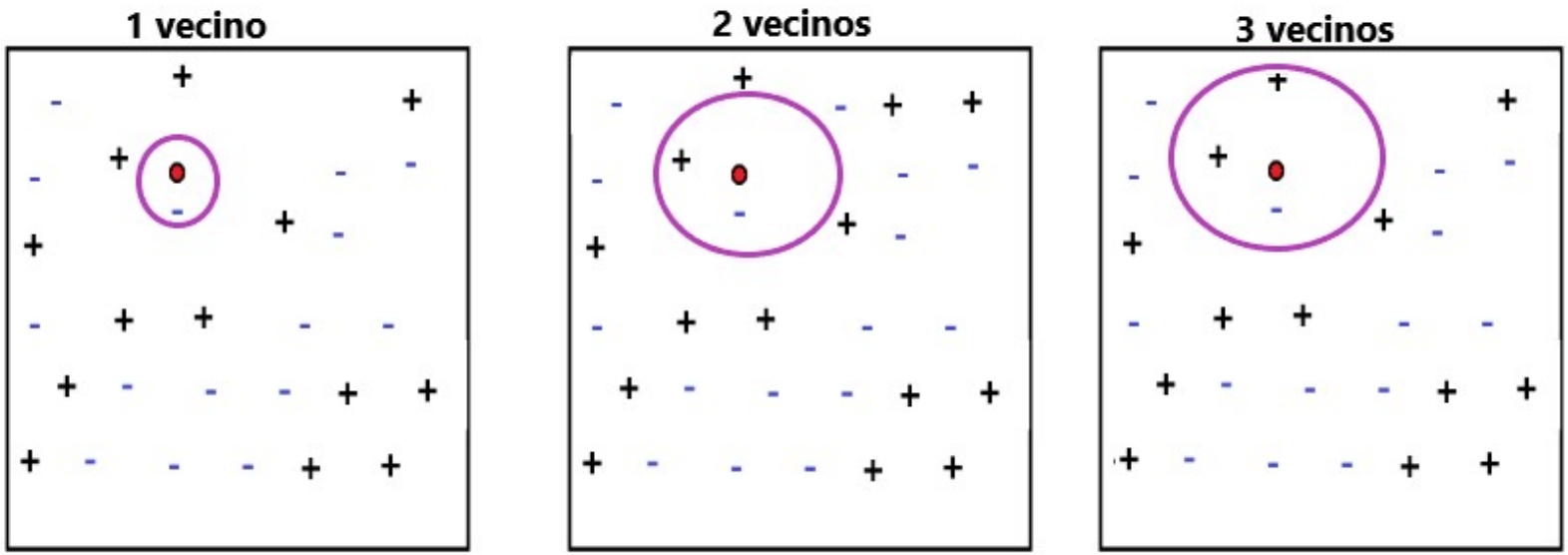
Miremos a los Vecinos más cercanos (kNN: k-nearest neighbors)



Miremos a los 3 Vecinos más cercanos (kNN: 3-nearest neighbors)



Miremos a los Vecinos más cercanos (kNN: k-nearest neighbors)



$k-$Vecinos más cercanos (kNN: k-nearest neighbors)

El método de $k-$Vecinos más cercanos es uno de los métodos existentes para estimar la distribución condicional de Y dado X y después clasificar una observación en la clase con la mayor probabilidad estimada.

- Elegimos k un entero positivo y un punto x para clasificar.
- Requiere una noción de distancia. Dados $\mathbf{z}, \mathbf{w} \in \mathbb{R}^q$

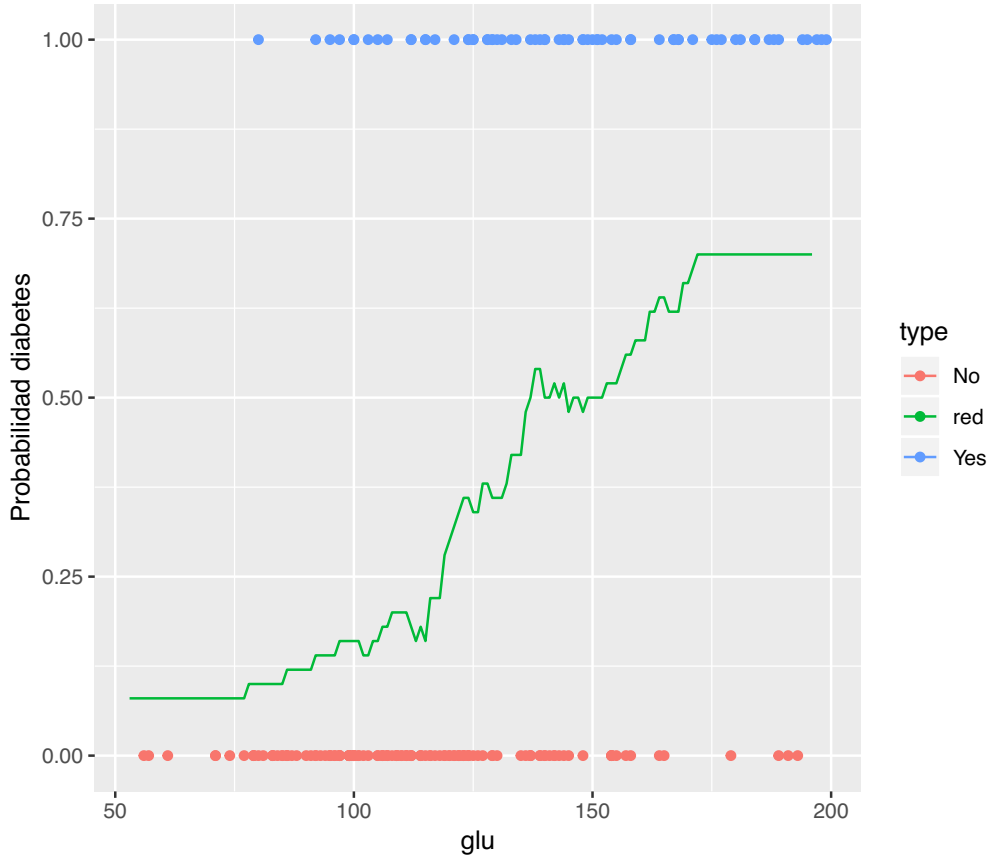
  Distancia euclídea: $d(\mathbf{z}, \mathbf{w}) = \sqrt{\sum_{i=1}^{q}(z_i - w_i)^2}$
- El clasificador kNN identifica el conjunto de los k puntos más cercanos a x. Sea N_x dicho conjunto.
- Estima a $P(Y = 1 \mid X = x)$ por la fracción de puntos en N_x cuya etiqueta es igual a 1:

$$\widehat{\mathbb{P}}(Y = 1 \mid X = x) = \frac{1}{k} \sum_{i \in N_x} \mathbb{I}(y_i = 1)$$

- El parámetro k de este método puede elegirse por Convalizadión Cruzada.

Ejemplo: Datos PIMA

```
> aa=ggplot(diabetes_train, aes(x = glu, y= as.numeric(type=='Yes'), colour = ty
+   geom_point() +
+   geom_line(data=datos_graf,aes(x=glu, y = probas,colour="red")) +
+   ylab('Probabilidad diabetes')+coord_fixed(ratio=150)
```



Algunas características

- Es un método muy intuitivo en el que a un nuevo punto se le asigna una categoría por voto de la mayoría entre los k vecinos más cercanos.
- Se generaliza muy fácilmente a una problema con más de dos clases.
- Se pueden usar distintas distancias.
- Si k es muy pequeño es muy sensible al ruido, si es muy grande podría incluir vecinos de otras clases.
- Para prevenir que alguno de los atributos tenga más influencia en la medida de distancia que otros se suele escalar, de esta manera una distancia d signifique lo mismo para el atributo 1 y para el 2, por ejemplo.
- Atributos irrelevantes podrían incrementar la distancia artificialmente a casos similares.
- Maldición de la dimensión.
- La clasificación de nuevos registros es más costosa que con otros métodos. Es un clasificador de aprendizaje perezoso (lazy).
- No construye un modelo explícito.

Otro Enfoque: Regresión Logística

La regla de óptima de Bayes depende de:

$$p(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})$$

$\mathbf{X} = (X_1, \ldots, X_p)^t$ es un vector de p covariables.

¿Y si modelamos esta probabilidad en función de $\mathbf{X}$?

Otro Enfoque: Regresión Logística

La regla de óptima de Bayes depende de:

$$p(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})$$

$\mathbf{X} = (X_1, \ldots, X_p)^t$ es un vector de p covariables.

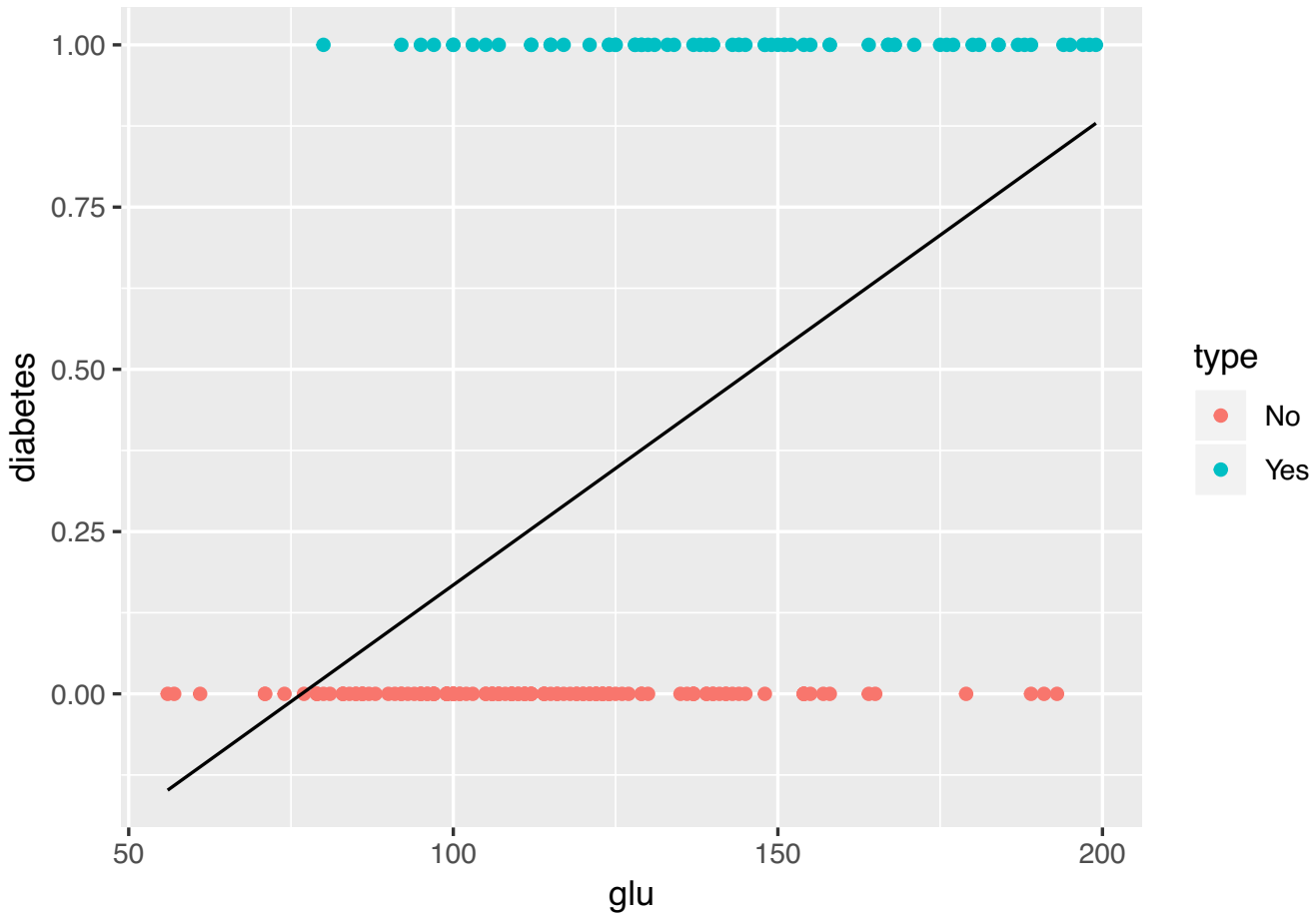
¿Y si modelamos esta probabilidad en función de $\mathbf{X}$?

Tenemos que tener algunos cuidados...

¿Qué pasa si proponemos un modelo lineal?

```
> datos_ent <- as.data.frame(MASS::Pima.tr)
> attach(datos_ent)
> diabetes<- 1*(type=="Yes")
> datos_ent$diabetes<- diabetes
> mean(diabetes)

[1] 0.34

> valor_new <- 70
> m_lineal <- lm(diabetes ~ glu, data = datos_ent)
> predict(m_lineal, data.frame(glu = valor_new))

          1
-0.04782927

>
```

¿Qué pasa si proponemos un modelo lineal?



Regresión Logística

Pensemos que tenemos una sola variable X

Notemos que

$$0 \quad \leq \quad p(x) \leq 1$$

Regresión Logística

Pensemos que tenemos una sola variable X

Notemos que

$$0 \quad \leq \quad p(x) \leq 1$$

Sin embargo, los **odds** (o chances en castellano...)

$$\frac{p(x)}{1-p(x)} \geq 0$$

Regresión Logística

Pensemos que tenemos una sola variable X

Notemos que

$$0 \quad \leq \quad p(x) \leq 1$$

Sin embargo, los **odds** (o chances en castellano...)

$$\frac{p(x)}{1-p(x)} \geq 0$$

Tomando logaritmo:

$$-\infty \quad < \quad \log\left(\frac{p(x)}{1-p(x)}\right) < \infty$$

Podríamos modelar:

$$\log\left(\frac{p(x)}{1-p(x)}\right) \quad = \quad \beta_0 + \beta_1\, x$$

Regresión Logística

Haciendo el camino inverso:

$$\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x}$$

$$p(x) = p(x,\beta) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

Regresión Logística

Haciendo el camino inverso:

$$\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1\, x}$$

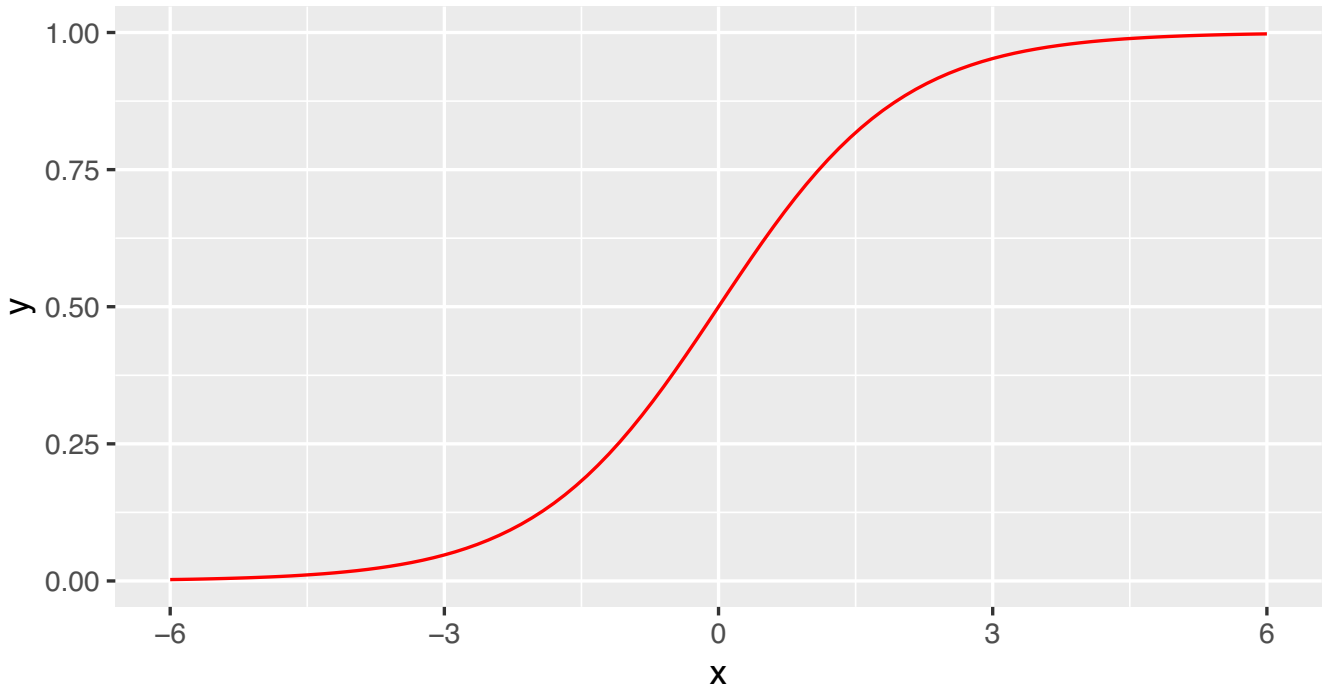
$$p(x) = p(x,\beta) = \frac{e^{\beta_0+\beta_1\, x}}{1+e^{\beta_0+\beta_1\, x}} = \frac{1}{1+e^{-\beta_0-\beta_1\, x}}$$

Función logística

La función logística está definida por

$$p(x) = \frac{e^x}{1 + e^x}$$

que corresponde en nuestro caso a tomar $\beta_0 = 0$ y $\beta_1 = 1$.



Regresión Logística

Con una sola variable

$$p(x) = p(x, \beta) \quad = \quad \frac{e^{\beta_0 + \beta_1 \, x}}{1 + e^{\beta_0 + \beta_1 \, x}}$$

Regresión Logística

Con una sola variable

$$p(x) = p(x, \beta) \quad = \quad \frac{e^{\beta_0 + \beta_1 \, x}}{1 + e^{\beta_0 + \beta_1 \, x}}$$

En general, tendremos para un vector de covariables

Modelo de regresión logística:

$$\mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}} \tag{1}$$

estimadores de máxima verosimilitud....

Estamos ante un problema un poco más complejo porque aquí las probabilidades están relacionadas a través de una función link con el parámetro β.

El EMV de $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ se halla maximizando

$$L(\mathbf{b}) = \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{b})^{y_i} (1 - p(\mathbf{x}_i, \mathbf{b}))^{1-y_i}$$

La log verosimilitud resulta

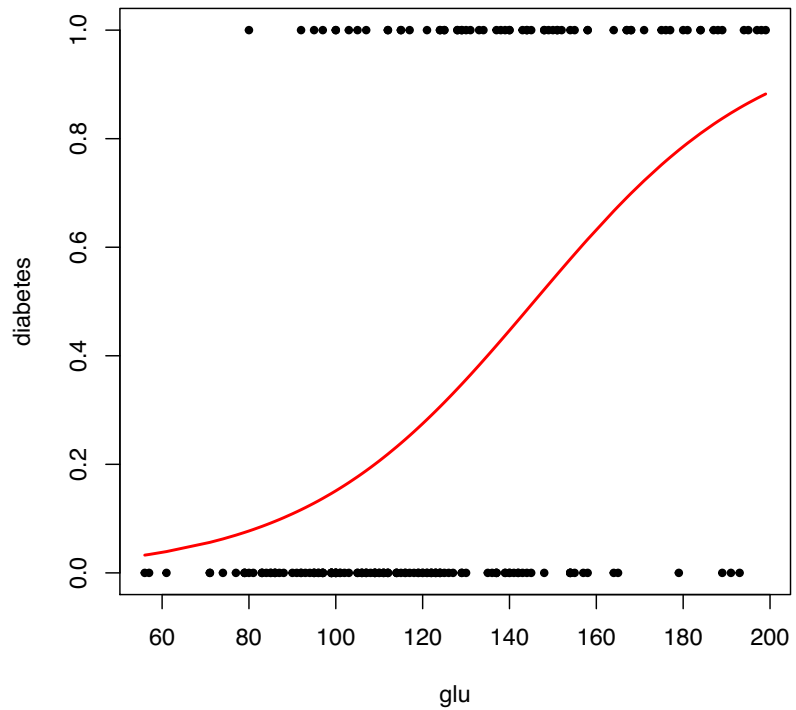
$$\ell(\mathbf{b}) = \log L(\mathbf{b}) = \sum_{i=1}^{n} y_i \log p(\mathbf{x}_i, \mathbf{b}) + (1 - y_i) \log(1 - p(\mathbf{x}_i, \mathbf{b}))$$

Típicamente, para hallar el EMV se deriva la log–verosimilitud $\ell(\boldsymbol{\beta})$ e iguala a 0:

$$\frac{\partial \ell}{\partial b_j} = \sum_{i=1}^{n} (y_i - p(\mathbf{x}_i, \mathbf{b})) x_{ij} = 0 \quad j = 0, 1 \ldots, p$$

Grafiquemos

```
> orden<- order(glu)
> plot(glu[orden],diabetes[orden],pch=20,xlab="glu",ylab="diabetes")
> lines(glu[orden],salida_glm$fitted.values[orden],col="red",lwd=2)
```



Grafiquemos con ggplot2

```
> library(ggplot2)
> aa<-ggplot(datos_ent, aes(x=glu, y=diabetes)) + geom_point() +
+   stat_smooth(method="glm", method.args=list(family="binomial"), se=F)
```