Descripción del problema

- Muestra de entrenamiento Tenemos $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathbb{E} \times \{1, \ldots, m\}$
  - X_i son características observadas y usualmente $\mathbb{E} = \mathbb{R}^d$
  - etiquetas indicando la naturaleza de las observaciones
- Muestra de Clasificación
  - características observadas, X.
  - etiquetas desconocidas.
- Problema: clasificar un nuevo dato del cual no se conoce la etiqueta.

El clasificador

- Dado $x \in \mathbb{E}$,
- queremos saber a que grupo $y \in \{1, ..., m\}$ pertenece.

Clasificador

$g : \mathbb{E} \to \{1, ..., m\}$

buscamos la función g que realice la asignación de las etiquetas.

Reglas de clasificación

- Análisis discriminante de Fisher.
- Modelos lineales generalizados.
- Vecinos más cercanos.
- CART.
- Redes Neuronales.

Error de Bayes

ERROR DE BAYES:
Probabilidad de caer en la banda que separa las poblaciones
Por la naturaleza de los datos, siempre esta presente
Es la cantidad total de observaciones mal clasificada
sin entrar en el detalle de cual es el error que se comete

Sea $(X, Y) \in \mathbb{E} \times \{1, ..., m\}$ un par aleatorio.
Se comete un error si $g(X) \neq Y$.
La probabilidad de clasificar erróneamente para g es

$$L(g) = P[g(X) \neq Y].$$

El mejor clasificador posible es la función g^* que minimiza $L(g)$.
Hay problemas fáciles y otros difíciles, por más que aumentemos el tamaño de la muestra los errores no van a bajar de determinada proporción.

Clasificadores basados en la muestra de entrenamiento

Consideramos la muestra de entrenamieto de pares independientes $\{(X_i, Y_i) : 1 \leq i \leq n\}$.
Un clasificador basado en la muestra de entrenamiento es

Muestra nueva que viene

$$g_n(\cdot; X_1, Y_1, ..., X_n, Y_n) : \mathbb{E} \to \{1, ..., m\}$$

El desempeño se mide por la probabilidad condicional de g_n de cometer un error.

$$L_n(g_n) = P\left[g_n(X; X_1, Y_1, ..., X_n, Y_n) \neq Y | X_1, Y_1, ..., X_n, Y_n\right].$$

Consistencia

Consideremos los siguientes mapeos

$$g_n : \mathbb{E} \times (\mathbb{E} \times \{1, ..., m\})^n \to \{1, ..., m\}$$

los llamamos reglas de clasificación. Una sucesión de clasificadores se denomina regla de clasificación $\{g_n : n \geq 1\}$.
Una regla es consistente si

$$\lim_n L_n(g_n) = L^*$$

.

El clasificador óptimo

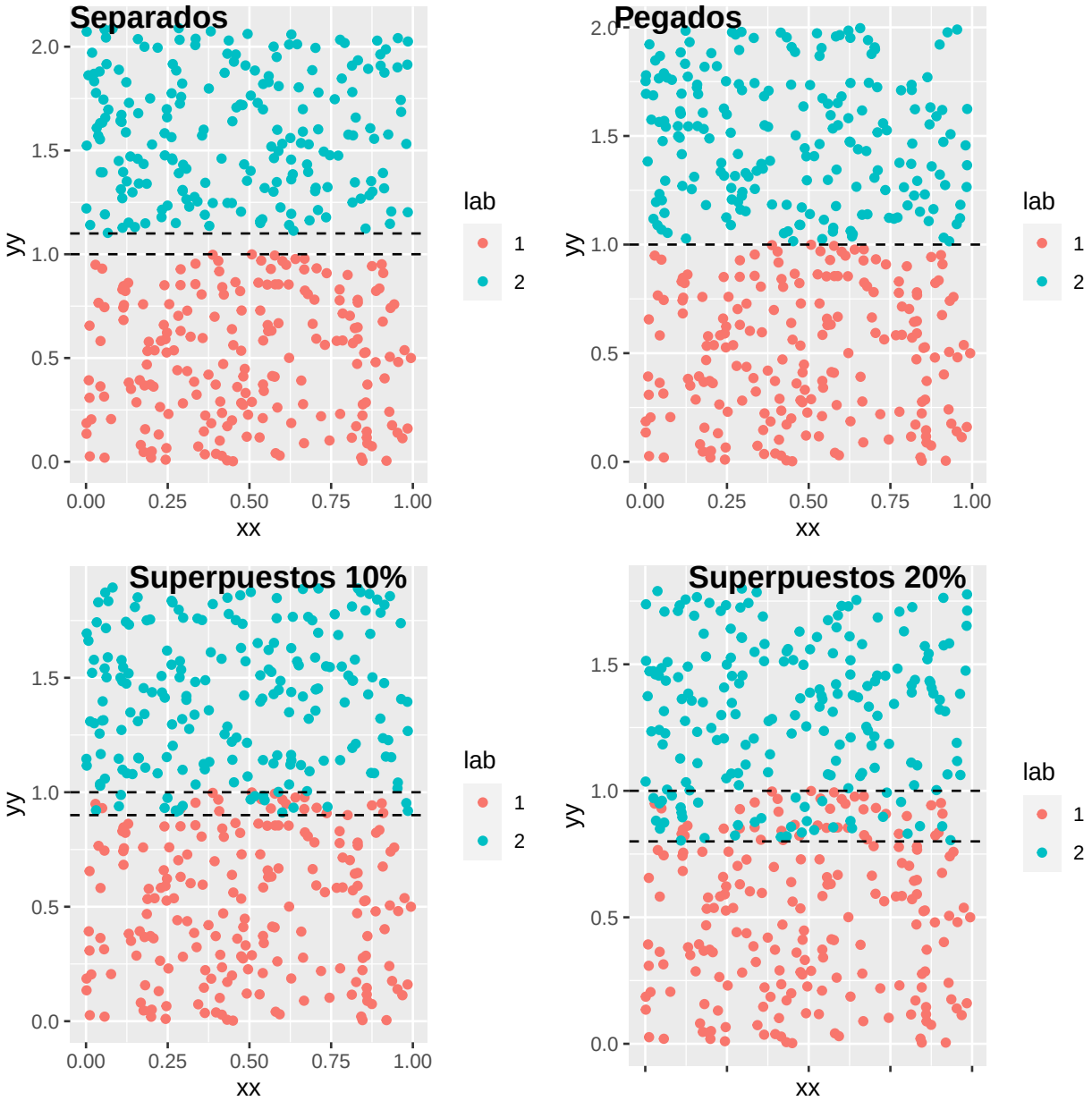
Interrogantes:

- Cómo se construye un buen clasificador?
- Cómo se mide el desempeño de un clasificador?
- Dados dos clasificadores A y B, como sabemos cuál es mejor?
- Cómo se estima el desempeño de un clasificador?
- Existe un clasidicador universamente óptimo?

El clasificador óptimo

Podemos considerar que una regla de clasificación es buena si es consistente $L_n(g_n) \to L^*$, en probabilidad.
Es decir que si aumentamos el tamaño de la muestra recontruiremos aproximadamente la distribución de (X, Y).
Una regla de clasificación se llama **universalmente consistente** si es consistente para cualquier distribución arbitraria de (X, Y).

El clasificador óptimo



El clasificador óptimo

En 1977 Stone, prueba que la regla de clasificación de k vecinos más cercanos es universalmente consistente, luego se obtienen otros resultados de convergencia universal para otras reglas de clasificación.
Cómo hacemos para medir cuán bueno es un clasificador?
Sabemos que $L_n \geq L^*$, sería bueno conococer $P(L_n \geq L^* + \epsilon)$.
Esta probabilidad depende de la distribución de (X, Y)!!!!!
No hay una regla de clasificación que tenga una tasa de convergencia universal.

There's no free lunch!

NOTA: Los modelos parametricos tienen una tasa de convergencia SQRT(n)

El clasificador óptimo

Si queremos estimar el desempeño de L_n para una regla específica, podríamos considerar por ejemplo,

$$\widehat{L}_n = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}_{\left\{g_{n_i}(X_i) \neq Y_i\right\}}$$

donde, $g_{n_i}(X_i)$ es la regla de clasificación construida en forma leave-one-out para cada observación (X_i, Y_i).
Estamos estimando el riesgo de Bayes, *riesgo empírico*.

El clasificador óptimo

Por simplicidad asumimos que tenemos dos categorías, $m = 2$. Contamos con la información dada por la muestra de entrenamiento,

$$\{(X_i, Y_i), i = 1, \ldots, n\}$$

- las observaciones $X_i \in \mathbb{E}$ son independientes.
- las Y_i toman el valor 0 ó 1 cuando el individuo pertenece a P_0 ó P_1 respectivamente.

las observaciones $X_i \in \mathbb{E}$ son independientes.

El clasificador óptimo

Queremos hallar una regla de clasificación que minimice el error de Bayes,

$$g : \mathbb{E} \to \{0, 1\} \text{ que minimice } P(g(X) \neq Y).$$

Se puede ver que está regla está dada por

$$g^*(X) = \mathcal{I}_{\{\eta(X)>1/2\}}, \quad \begin{cases} 1 & n(x) > 1/2 \\ 0 & n(x) <= 1/2 \end{cases}$$

donde $\eta(X) = E(Y|X)$, luego tenemos que estimar la esperanza condiciona de Y dado X. Si X es discreta es sencillo de estimar, sino es más complicado y lo analizaremos a lo largo del curso.

Regla de Fisher

El clasificador lineal de Fisher es una de las reglas más sencillas y su uso está altamente difundido.
Supongamos que tenemos dos poblaciones con distribuciones P_0 y P_1, que tienen igual varianza Σ pero distinta media μ_0 y μ_1.
Idea:
encontrar una transformación lineal de las variables, $X \to \beta' X$ de forma tal que x se asigna a la población P_0 si
$\beta' X$ está *más cerca* de $\beta' \mu_0$ que de $\beta' \mu_1$.

Regla de Fisher

La transformación lineal se elige maximizando (en β) la separación entre las medias proyectadas,

$$\beta'\mu_0 \text{ y } \beta'\mu_1.$$

Para que el problema anterior tenga solución, la maximización se hace sujeta a la restricción que la varianza común $\beta'\Sigma\beta$ de $\beta'X$ sea 1. Esto reduce el problema a maximizar en β el siguiente cociente entre *varianza entre clases* y la *varianza dentro de clases*,

$$\frac{(\beta'\mu_0 - \beta'\mu_1)^2}{\beta'\Sigma\beta}.$$

Regla de Fisher

El clasificador que resulta es equivalente a asignar una nueva observación x minimizando la distancia de Mahalanobis a las medias μ_0 y μ_1, x se asigna a P_0 si

$$(x - \mu_0)'\Sigma(x - \mu_0) < (x - \mu_1)'\Sigma(x - \mu_1)$$

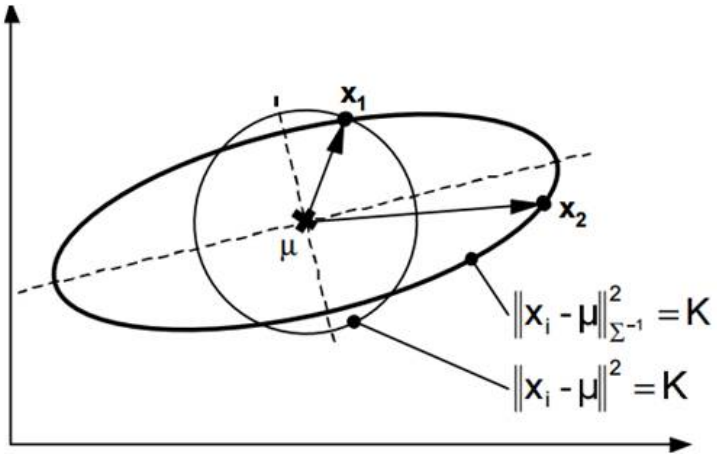
La distancia de Mahalanobis

Toma la forma de la elipse porque se lo divide por la escala

La distancia de Mahalanobis

Mahalanobis Distance

$$\|x - y\|^2_{\Sigma_i^{-1}} = (x-y)^T \Sigma_i^{-1} (x-y)$$

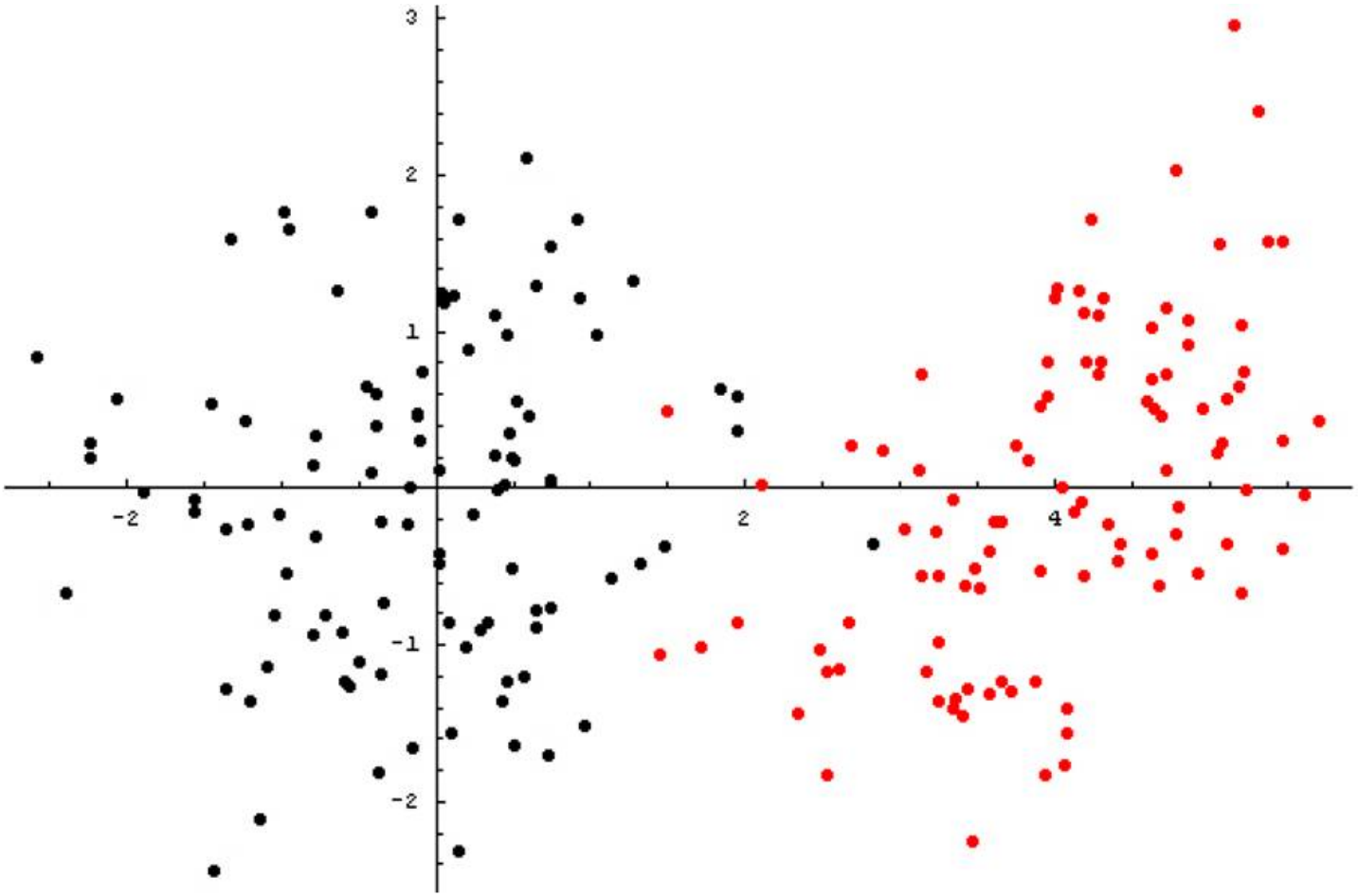


- En $\mathbb{R}$ es medir la distancia en cantidad de desvíos estándares.
- Si la matriz de covarianza es la identidad entonces la distancia Euclidea y de Mahalanobis coinciden.

Distancia de Mahalanobis

Datos $n = 100$ donde $P_1 \sim N(\mu_1, \Sigma_1)$ y $P_2 \sim N(\mu_2, \Sigma_2)$.

$\mu_1 = (0,0)', \mu_2 = (4,0)', \Sigma_1 = Id$ y $\Sigma_2 = \begin{bmatrix} 1 & 1/\sqrt{3} \\ 1/\sqrt{3} & 1 \end{bmatrix}$

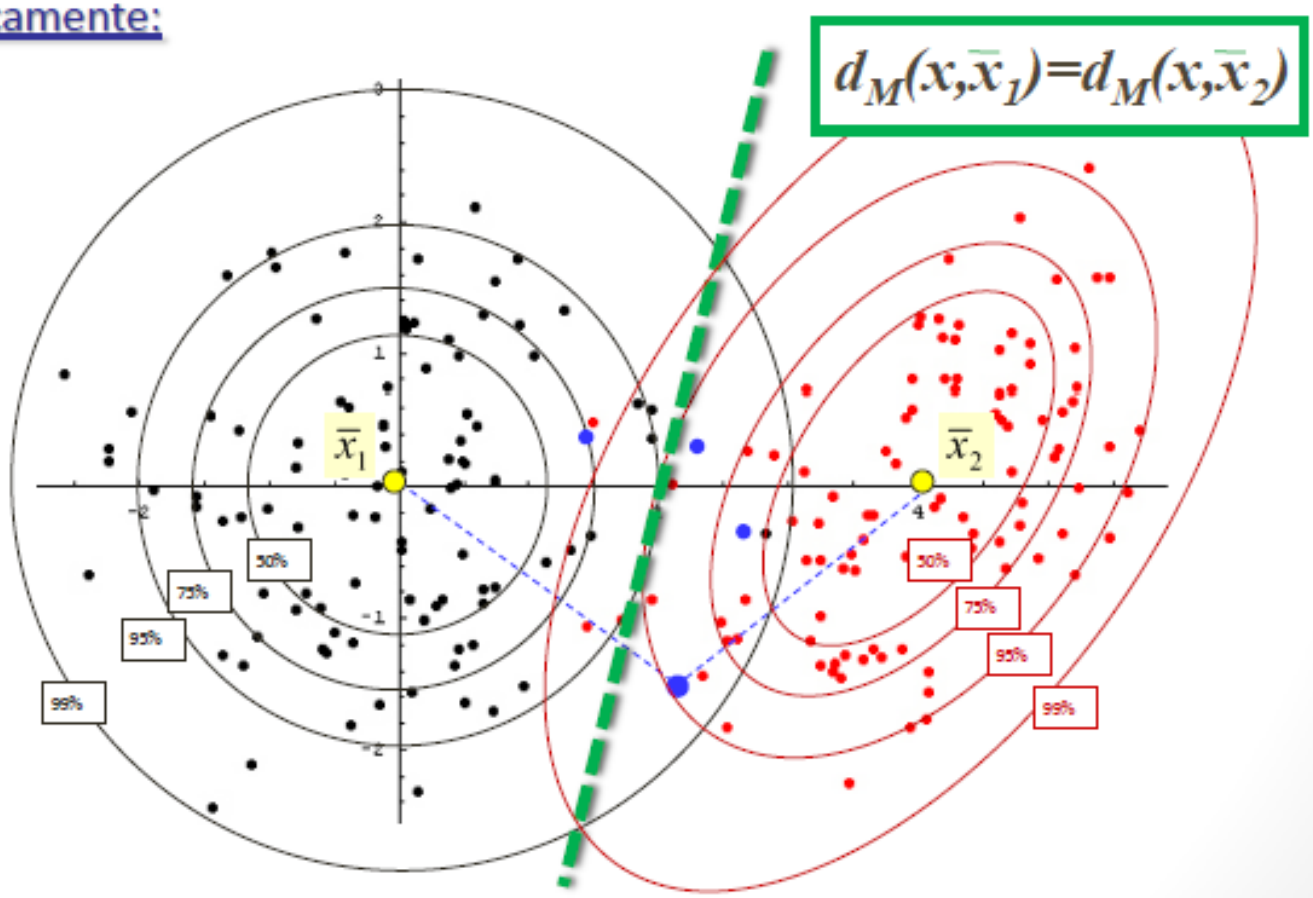


Distancia de Mahalanobis

Datos $n = 100$ donde $P_1 \sim N(\mu_1, \Sigma_1)$ y $P_2 \sim N(\mu_2, \Sigma_2)$.

$\mu_1 = (0,0)', \mu_2 = (4,0)', \Sigma_1 = Id$ y $\Sigma_2 = \begin{bmatrix} 1 & 1/\sqrt{3} \\ 1/\sqrt{3} & 1 \end{bmatrix}$

Gráficamente:

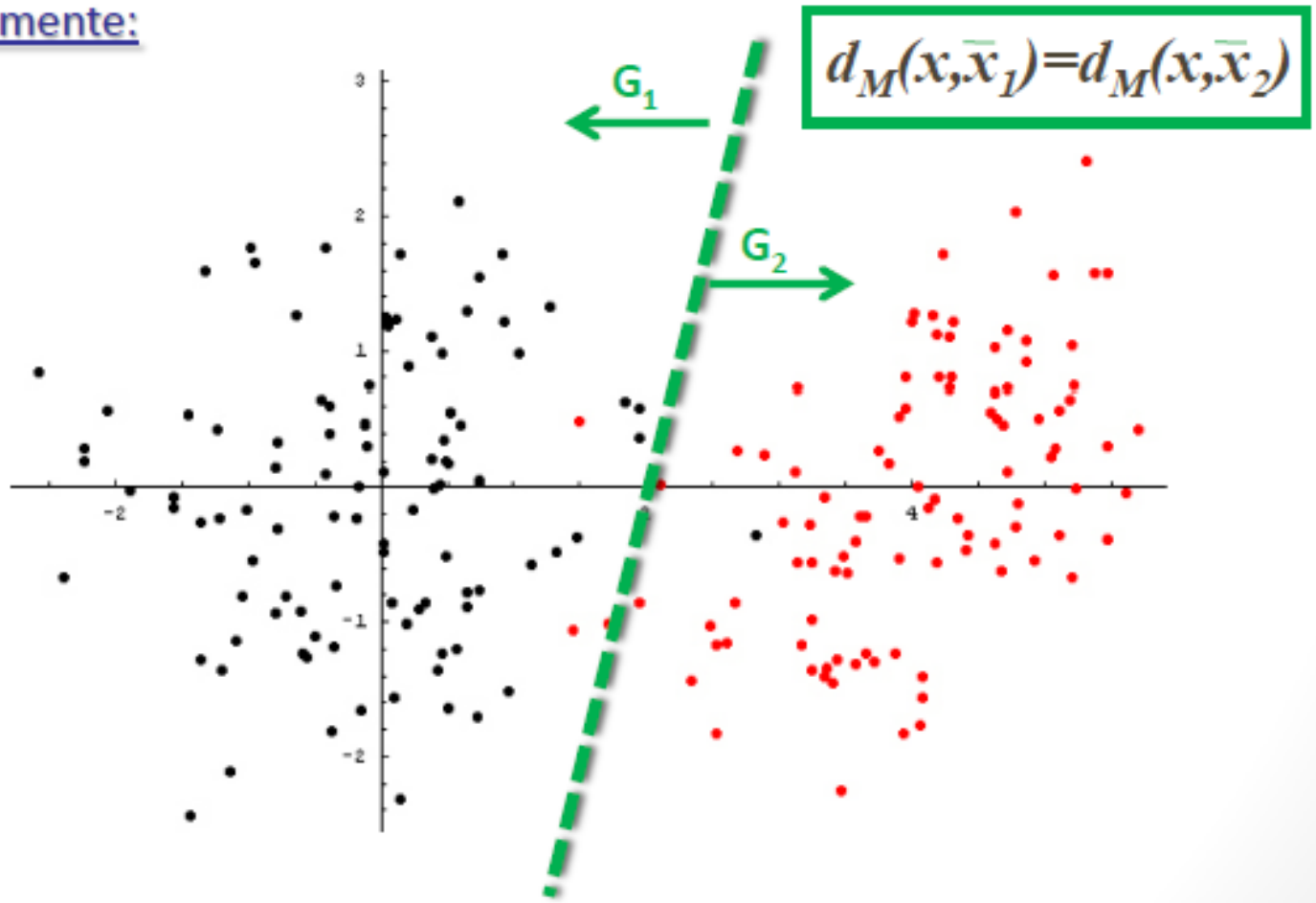


Distancia de Mahalanobis

Datos $n = 100$ donde $P_1 \sim N(\mu_1, \Sigma_1)$ y $P_2 \sim N(\mu_2, \Sigma_2)$.

$\mu_1 = (0,0)', \mu_2 = (4,0)', \Sigma_1 = Id$ y $\Sigma_2 = \begin{bmatrix} 1 & 1/\sqrt{3} \\ 1/\sqrt{3} & 1 \end{bmatrix}$

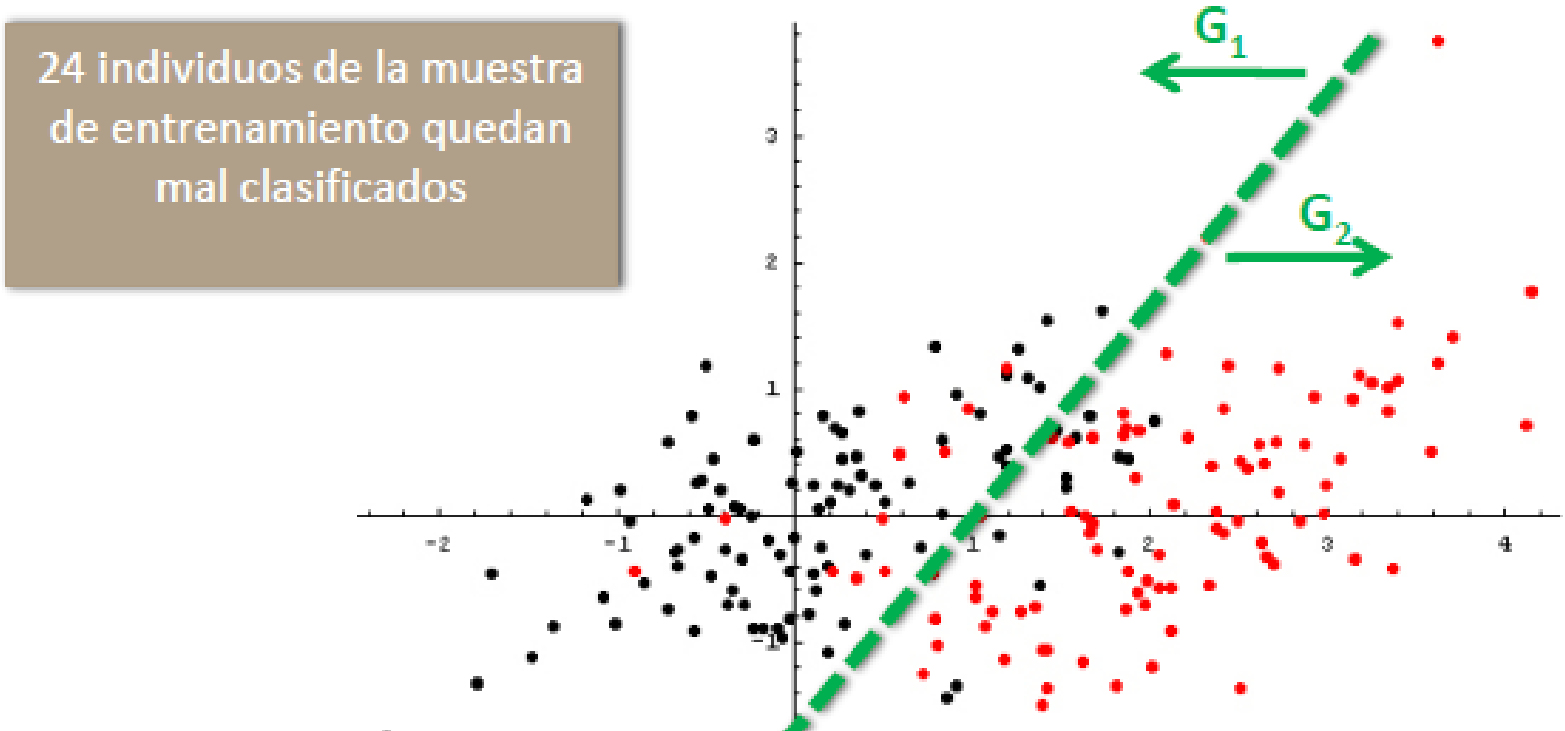
Gráficamente:



Distancia de Mahalanobis

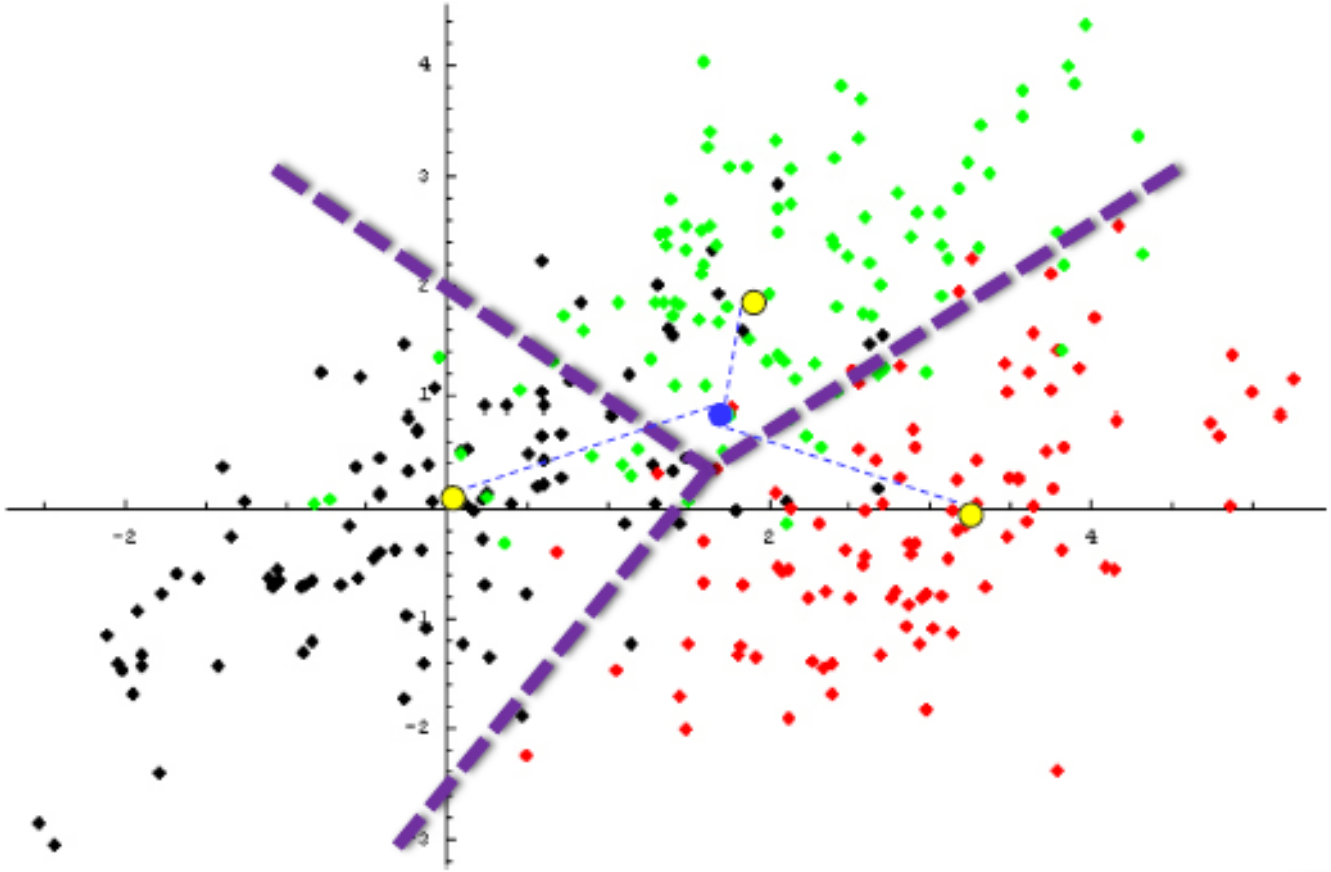
Para que el método de clasificación de Mahalanobis funcione bien:

- las distribuciones de los datos tienen que ser aproximadamente elíticas.
- las variable utilizadas tienen que ser relevantes a los fines de la clasificación.



Distancia de Mahalanobis

Cuando hay más de dos grupos se procede en forma análoga, calculando para cada punto a clasificar la distancia de Mahalanobis al centro de cada grupo y clasificándolo en el que tenga menor distancia.



Regla de Fisher: Análisis Discriminante Lineal

- Consideramos 2 poblaciones P_0 y P_1.
- Sea $\mathbf{x}_0$ un vector aleatorio de dimensión p definido para esas poblaciones.
- Supongamos que f_0 y f_1 son las densidades conocidas.

$$\pi_i = \text{ probabilidad de pertenecer a la población } i.$$

- Luego $f(\mathbf{x}_0) = \pi_0 f_0(\mathbf{x}_0) + \pi_1 f_1(\mathbf{x}_0)$.

Regla de Fisher: Análisis Discriminante Lineal

- Sea $\mathbf{x}_0$ una observación a clasificar, entonces

$$P\left(0\,|\mathbf{x}_0\right) = \frac{P\left(\mathbf{x}_0\,|0\right)\pi_0}{P\left(\mathbf{x}_0\,|0\right)\pi_0 + P\left(\mathbf{x}_0\,|1\right)\pi_1}.$$

- Teniendo en cuenta que $P\left(\mathbf{x}_0\,|0\right) = f_0\left(\mathbf{x}_0\right)\Delta\left(\mathbf{x}_0\right)$

$$\boxed{P\left(0\,|\mathbf{x}_0\right) = \frac{f_0\left(\mathbf{x}_0\right)\pi_0}{f_0\left(\mathbf{x}_0\right)\pi_0 + f_1\left(\mathbf{x}_0\right)\pi_1}.}$$

y análogo para P_1.

Regla de Fisher: Análisis Discriminante Lineal

Entonces clasificamos a $\mathbf{x}_0$ donde resulte más probale es decir, $\mathbf{x}_0 \in P_1$ si

Regla de clasificación

$$f_1\left(\mathbf{x}_0\right)\pi_1 > f_0\left(\mathbf{x}_0\right)\pi_0.$$

Regla de Fisher: Análisis Discriminante Lineal

Si pudieramos asignar un costo a cometer errores en la clasificación

$$c\left(1\left|0\right.\right) = \text{ costo de clasificar a } \mathbf{x}_0 \text{ en } P_1 \text{ cuando pertenece a } P_0.$$

Definimos $d_1 =$ clasificar a $\mathbf{x}_0$ en P_1. Entonces,

$$\begin{aligned} E\left(d_1\right) &= c\left(1\left|0\right.\right) P\left(0\left|\mathbf{x}_0\right.\right) + 0P\left(1\left|\mathbf{x}_0\right.\right) = c\left(1\left|0\right.\right) P\left(0\left|\mathbf{x}_0\right.\right) \\ &= c\left(1\left|0\right.\right) \frac{f_0\left(\mathbf{x}_0\right)\pi_0}{f_0\left(\mathbf{x}_0\right)\pi_0 + f_1\left(\mathbf{x}_0\right)\pi_1}. \end{aligned}$$

Costo, por clasificar mal

Regla de Fisher: Análisis Discriminante Lineal

Luego asignamos a $\mathbf{x}_0$ a la población con menor costo esperado. Es decir clasificamos a $\mathbf{x}_0$ en P_1 si

$$\frac{f_1(\mathbf{x}_0)\,\pi_1}{c(1\,|0)} > \frac{f_0(\mathbf{x}_0)\,\pi_0}{c(0\,|1)}.$$

Clasificamos a $\mathbf{x}_0$ en P_1, si

- su probabilidad a priori es más alta.
- la verosimilitud de que provenga de P_1 es más alta.
- el costo de equivocarnos al clasificarlo en P_1 es más bajo.

Regla de Fisher: Análisis Discriminante Lineal

Asumiendo distribución normal en los datos, tenemos que,

$$
\begin{aligned}
f_i(\mathbf{x_0}) &= \frac{1}{(2\pi)^{p/2}\,|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x_0}-\mu_i)'\Sigma^{-1}(\mathbf{x_0}-\mu_i)\right). \\
&= \frac{1}{(2\pi)^{p/2}\,|\Sigma|^{1/2}} exp\left(-\frac{1}{2}D_i^2\right).
\end{aligned}
$$

Donde D_i^2 es la distancia de Mahalanobis al cuadrado.

Regla de Fisher: Análisis Discriminante Lineal

Clasificamos a $\mathbf{x}_0$ en P_1

$$\frac{D_0^2}{2} - \log \frac{\pi_0}{c\,(0\,|1\,)} > \frac{D_1^2}{2} - \log \frac{\pi_1}{c\,(1\,|0\,)},$$

donde $D_i^2 = (\mathbf{x}_0 - \mu_i)' \, \Sigma^{-1} \, (\mathbf{x}_0 - \mu_i)$ es la distancia de Mahalanobis al cuadrado.
Si además $\pi_0 = \pi_1$ y además $c\,(0\,|1\,) = c\,(1\,|0\,)$, queda

Regla de clasificación

$$D_0^2 > D_1^2.$$

Regla de Fisher: Análisis Discriminante Lineal

Esta regla de clasificación la podemos ver como una función lineal en $\mathbf{x_0}$ se clasifica a $\mathbf{x}_0$ donde sea mínima. Recordemos

$$D_0^2 - \log \frac{\pi_0}{c\,(0\,|1\,)} > D_1^2 - \log \frac{\pi_1}{c\,(1\,|0\,)}$$

La frontera está dada por

$$D_0^2 - \log \frac{\pi_0}{c\,(0\,|1\,)} = D_1^2 - \log \frac{\pi_1}{c\,(1\,|0\,)}$$

despejando

$$(\mu_0 - \mu_1)'\,\Sigma^{-1}\mathbf{x_0} = (\mu_0 - \mu_1)'\,\Sigma^{-1}\left(\frac{\mu_0 + \mu_1}{2}\right) - \log \frac{\pi_1 c\,(0\,|1\,)}{\pi_0 c\,(1\,|0\,)},$$

nuevamente si $\pi_0 = \pi_1$ y además $c\,(0\,|1\,) = c\,(1\,|0\,)$, y llamamos $\mathbf{w} = (\mu_0 - \mu_1)'\,\Sigma^{-1}$.

$$\mathbf{w}'\mathbf{x_0} = \mathbf{w}'\left(\frac{\mu_0 + \mu_1}{2}\right).$$

Regla de Fisher: Análisis Discriminante Lineal

Si las poblaciones son normales.
Es decir que $\mathbf{x}_0$ lo clasificaremos en P_1 si

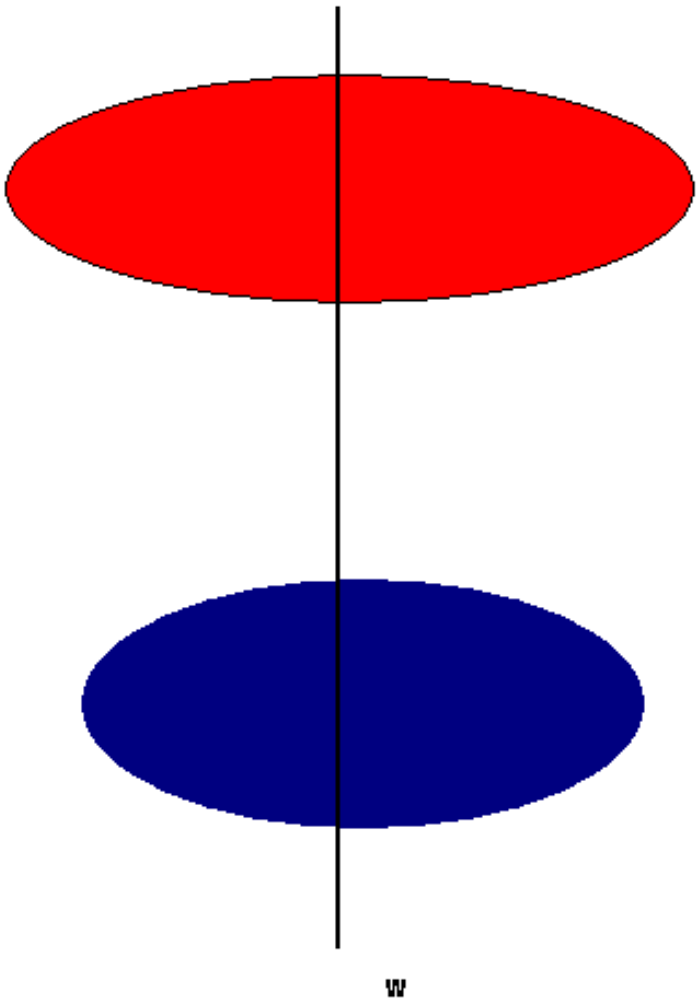
$$\mathbf{w}'\mathbf{x_0} > \mathbf{w}'\left(\frac{\mu_0 + \mu_1}{2}\right).$$

Regla de Fisher: Análisis Discriminante Lineal

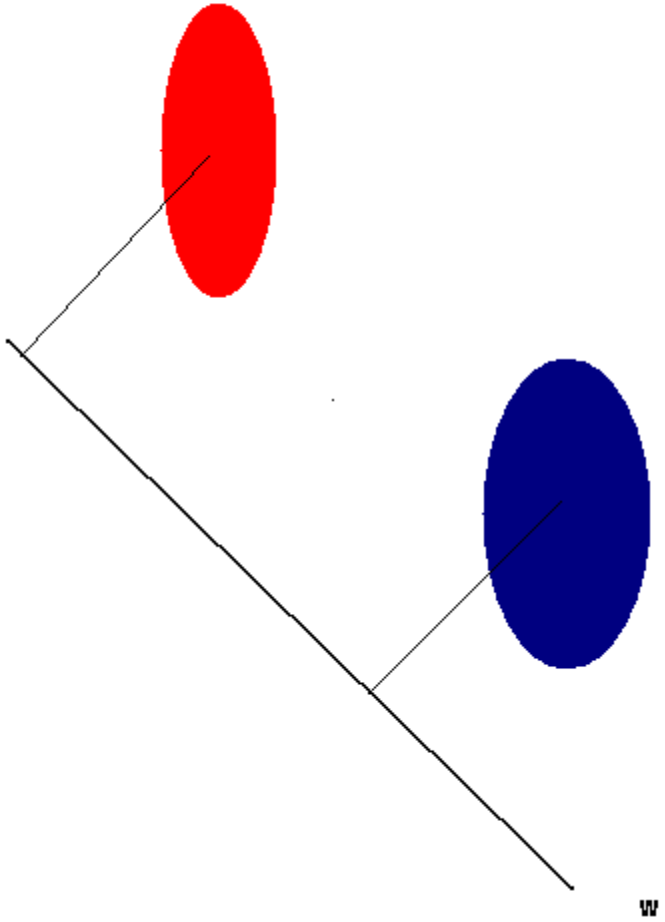
Procedimiento:

1. Construir $\mathbf{w} = (\mu_0 - \mu_1)' \Sigma^{-1}$. **"Direccion que mejor me separa las observaciones"**
2. $\mathbf{z}_0 = \mathbf{w}'\mathbf{x}_0$, proyecto la observación en la dirección de $\mathbf{w}$.
3. Clasificar a $\mathbf{z}_0$ en la población que tenga centro "proyectado" $\left(\mathbf{m}_i = \mathbf{w}' \mu_i\right)$ más cercano.

Interpretación Geométrica



Interpretación Geométrica



Regla de Fisher: Análisis Discriminante Lineal

Ahora consideramos que tenemos una muestra aleatoria de observaciones provenientes de P_0; $\mathbf{x}_{01}, \ldots, \mathbf{x}_{0n}$ y otra $\mathbf{x}_{11}, \ldots, \mathbf{x}_{1m}$ proveniente de P_1.
Estimamos

$$
\begin{aligned}
\widehat{\mu}_0 &= \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{0i}. \\
\widehat{\mu}_1 &= \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_{1i}. \\
\widehat{S}_0 &= \frac{1}{n-1} \sum_{i=1}^{n} \left(\mathbf{x}_{0i} - \widehat{\mu}_0\right)\left(\mathbf{x}_{0i} - \widehat{\mu}_0\right)'. \\
\widehat{S}_1 &= \frac{1}{m-1} \sum_{i=1}^{m} \left(\mathbf{x}_{1i} - \widehat{\mu}_1\right)\left(\mathbf{x}_{1i} - \widehat{\mu}_1\right). \\
\widehat{S}_w &= \frac{n-1}{n+m-2} \widehat{S}_0 + \frac{m-1}{n+m-2} \widehat{S}_1
\end{aligned}
$$

Luego clasificamos a la nueva observación $\mathbf{x}_0$ en el centro más próximo estimado, considerando la distancia relativa a la matriz de covarianza estimada.

Variables canónicas discriminantes

Encontrar una variable escalar

$$z = \alpha' \mathbf{x},$$

que maximice la distancia entre las medias proyectadas en relación a la variabilidad resultante de la proyección.
La media de P_0 proyectada en la dirección α, es

$$m_0 = \alpha' \mu_0$$

y su estimador

$$\widehat{m}_0 = \alpha' \widehat{\mu}_0.$$

De manera análoga para P_1

$$m_1 = \alpha' \mu_1$$

y su estimador

$$\widehat{m}_1 = \alpha' \widehat{\mu}_1.$$

Variables canónicas discriminantes

Consideramos que P_0 y P_1 tienen la misma varianza Σ, que la estimamos por $\widehat{S}_w$, luego

$$s_z^2 = \alpha' S_w \alpha.$$

Variables canónicas discriminantes

Se busca α de manera que la separación entre las medias se máxima, la versión empírica adimensional de la separación es

$$\phi = \left(\frac{\widehat{m}_0 - \widehat{m}_1}{s_z} \right)^2 .$$

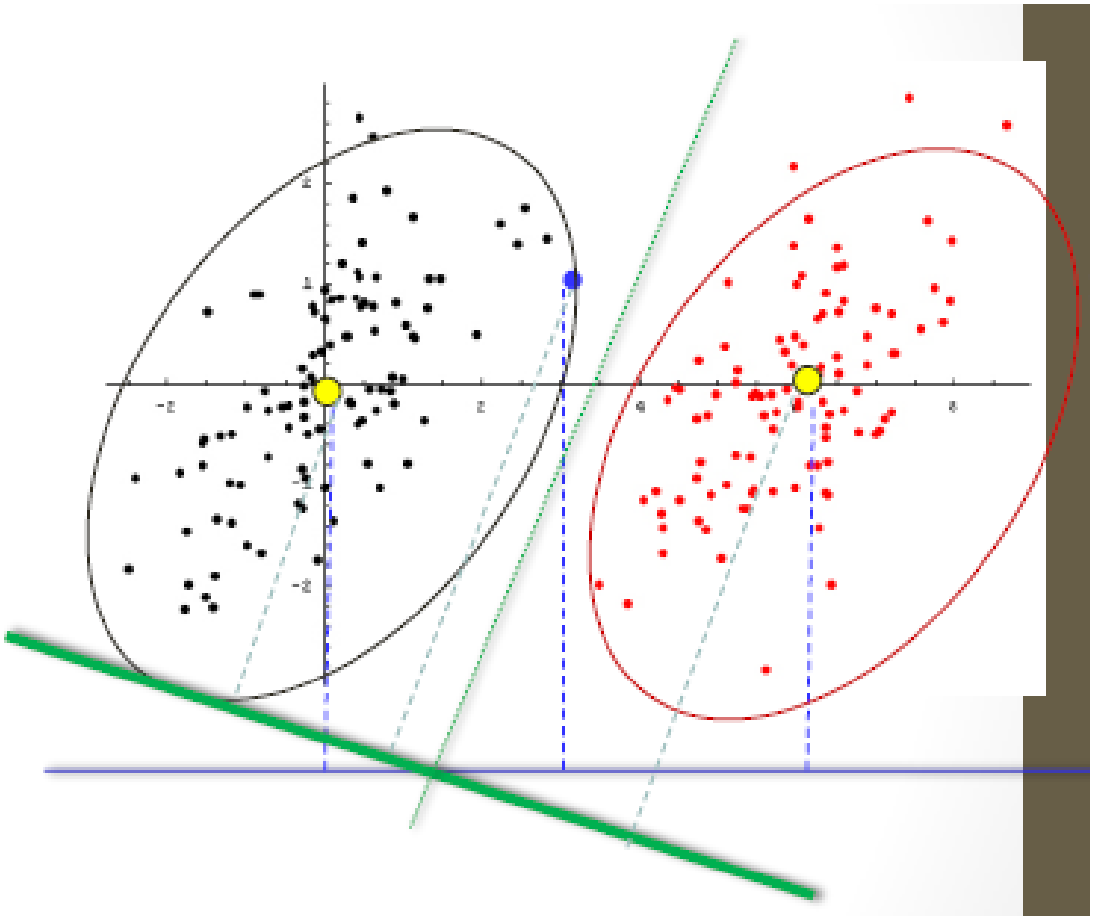
Regla de Fisher: Análisis Discriminante Lineal

Geométricamente Se busca una buena dirección sobre la que proyectar los datos de los grupos conocidos y de los que queremos clasificar. Se clasifica en función de qué grupo está más cerca en esa dirección. Una buena dirección

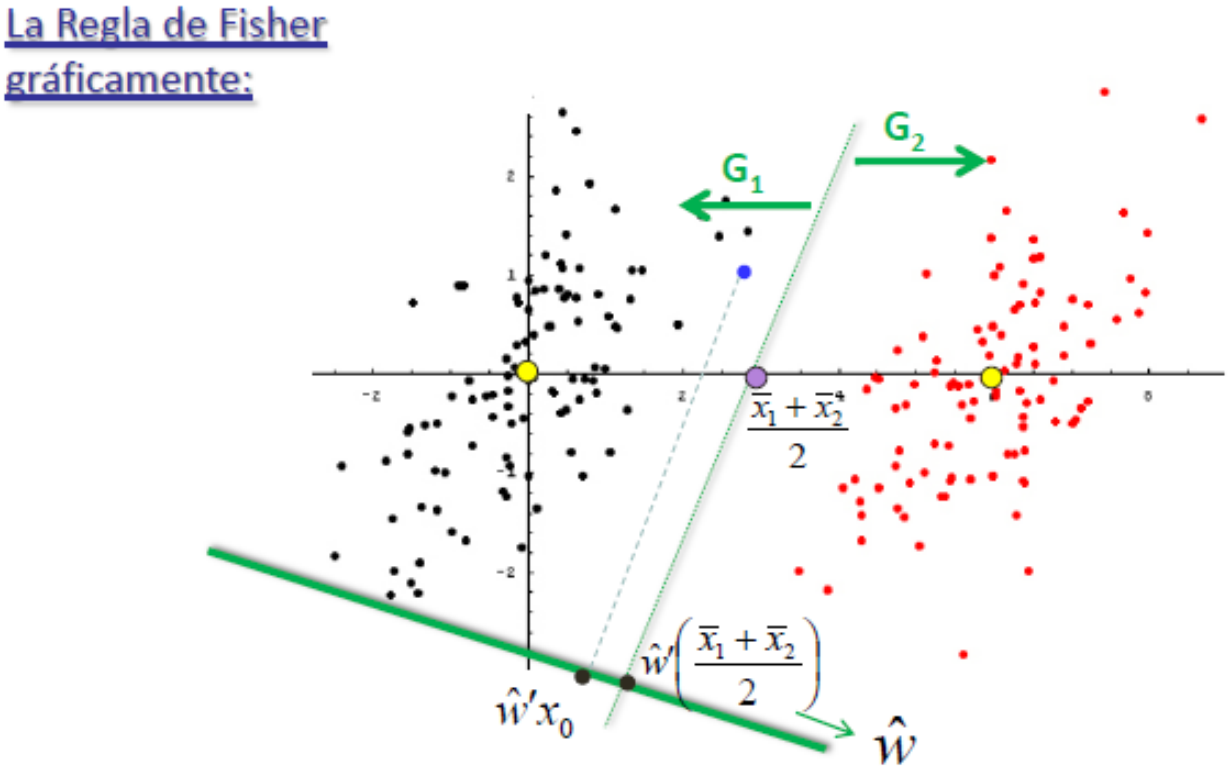
- separa las medias.
- tiene en cuenta la dispersión.

El nuevo dato se clasifica dentro del grupo con la media más próxima en la proyección.

Regla de Fisher: Análisis Discriminante Lineal



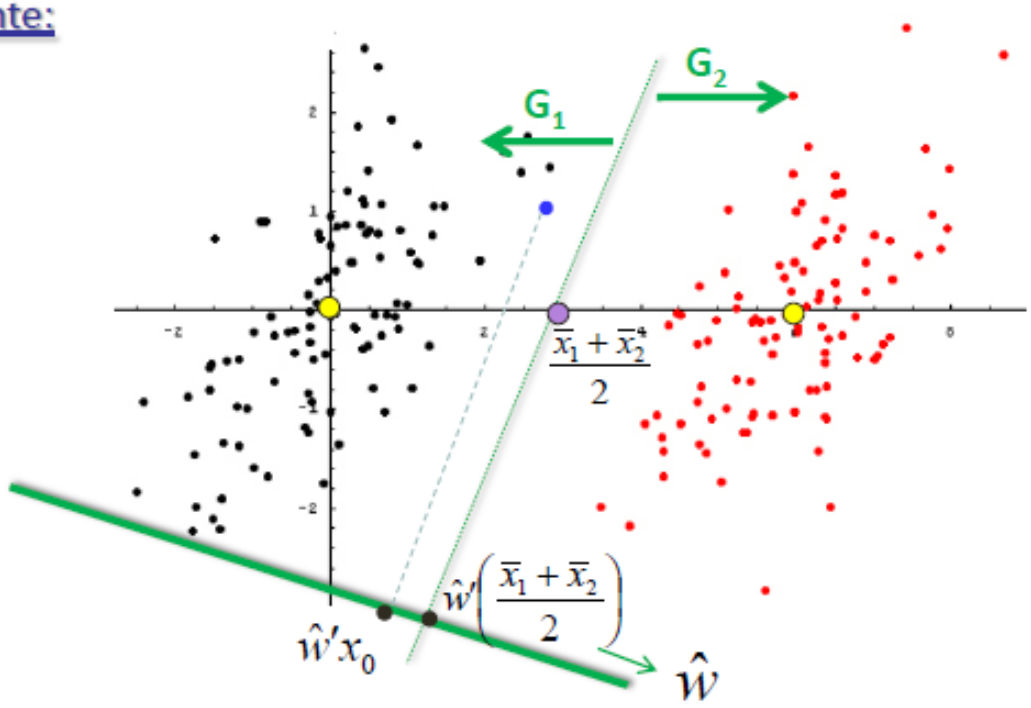
Regla de Fisher: Análisis Discriminante Lineal



Observación: en este caso, donde la matriz de covarianza coincide, los enfoques de Fisher y Mahalanobis dan las mismas reglas de clasificación.

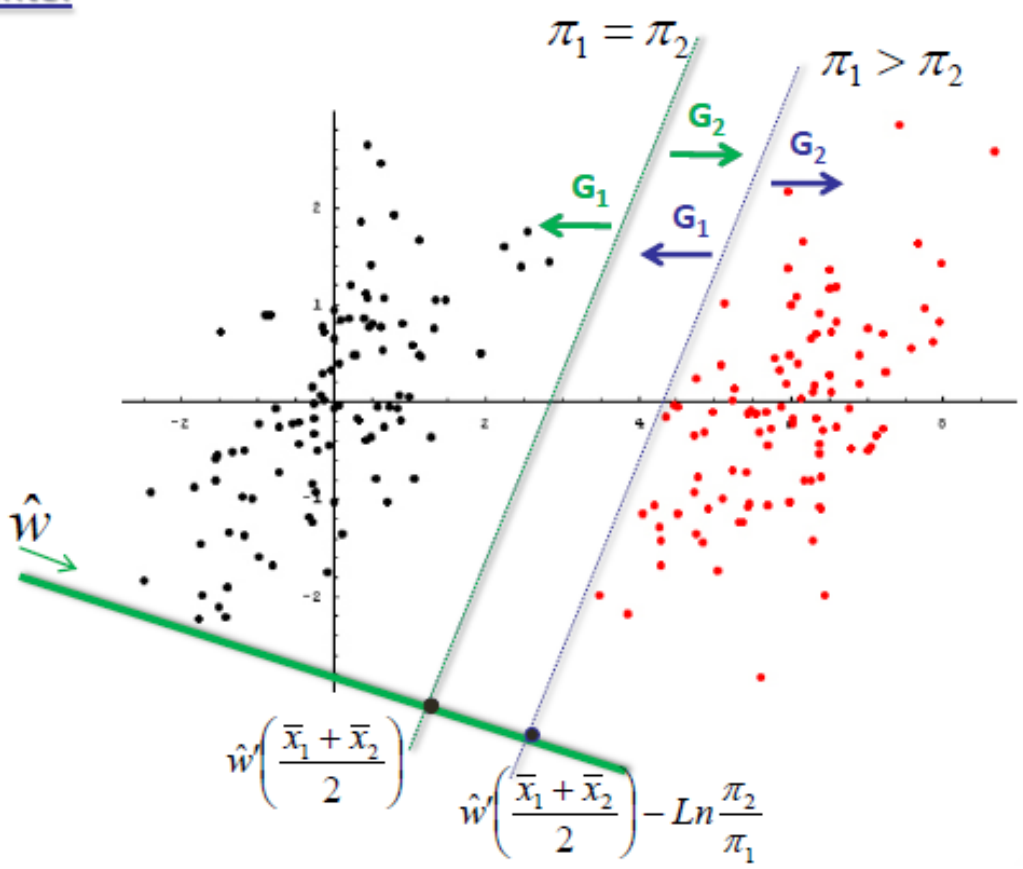
Regla de Fisher: Análisis Discriminante Lineal

La Regla de Fisher gráficamente:



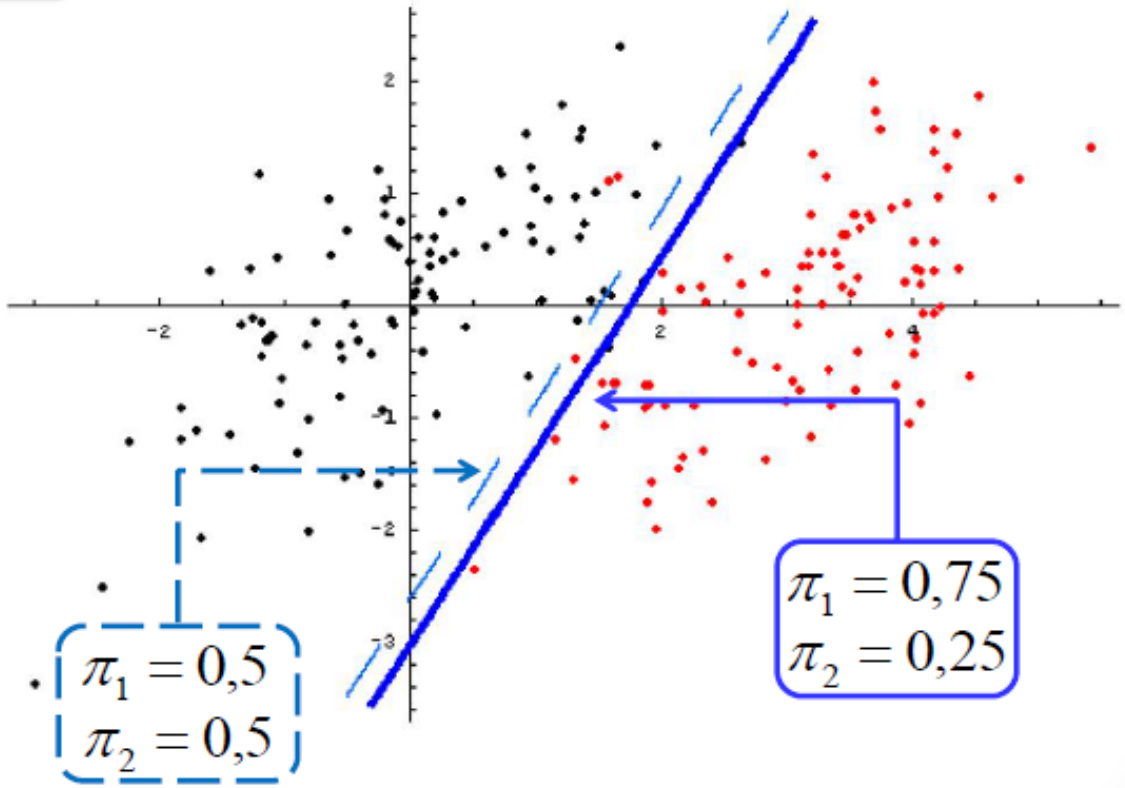
Regla de Fisher: Análisis Discriminante Lineal

Gráficamente:



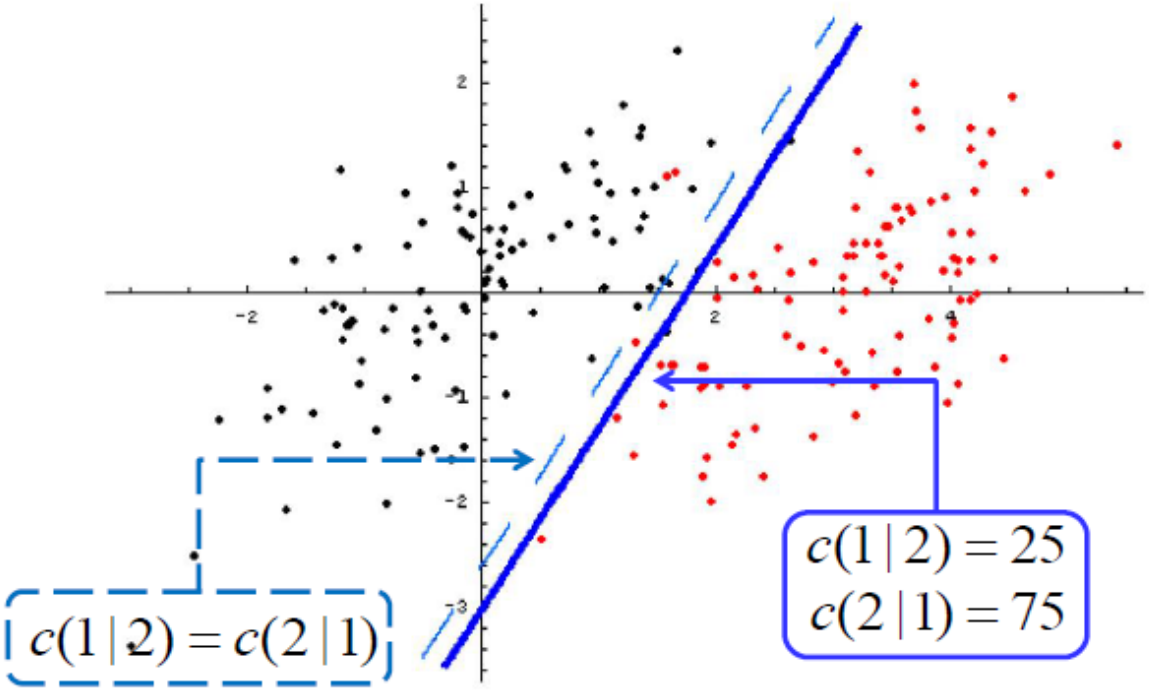
Regla de Fisher: Análisis Discriminante Lineal

Ejemplo:



Regla de Fisher: Análisis Discriminante Lineal

Ejemplo: $\pi_1 = \pi_2 = 0{,}5$



Análisis Discriminante lineal

Ejemplo: **Lirios.**

En este problema hay tres grupos de lirios, que queremos clasificar con 50 observaciones cada uno, así que tendremos que combinar dos reglas discriminantes para clasificar por especies a los nuevos ejemplares de lirios.

Setosa

Virginica

Versicolor

Variables:

SL	longitud del sépalo
SW	ancho del sépalo
PL	longitud del pétalo
PW	ancho del pétalo

Análisis Discriminante lineal

Selecciono las observaciones, cuales iran a cada muestra, el 75% a la muestra de entrenamiento.

```
training_sample = sample(c(TRUE, FALSE), nrow(iris),
replace = T, prob = c(0.75,0.25))
```

Tomo la muestra de entrenamiento

```
train = iris[training_sample, ]
```

Tomo la muestra de entrenamiento de testeo

```
test = iris[!training_sample, ]
```

Análisis Discriminante lineal

Linear discriminal analysis

Hago el análisis discriminante lineal, no estoy poniendo priors, en este caso podría poner 1/3 a cada una.

```
lda.iris = lda(Species ~., train)
lda.iris
```

Muestra la media de cada clase

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5.02	3.45	1.47	0.26
versicolor	5.89	2.75	4.27	1.31
virginica	6.66	3.04	5.62	2.09

Análisis Discriminante lineal

	LD1	LD2
Sepal.Length	0.86	-0.70
Sepal.Width	1.25	-1.55
Petal.Length	-2.13	1.70
Petal.Width	-2.70	-3.81

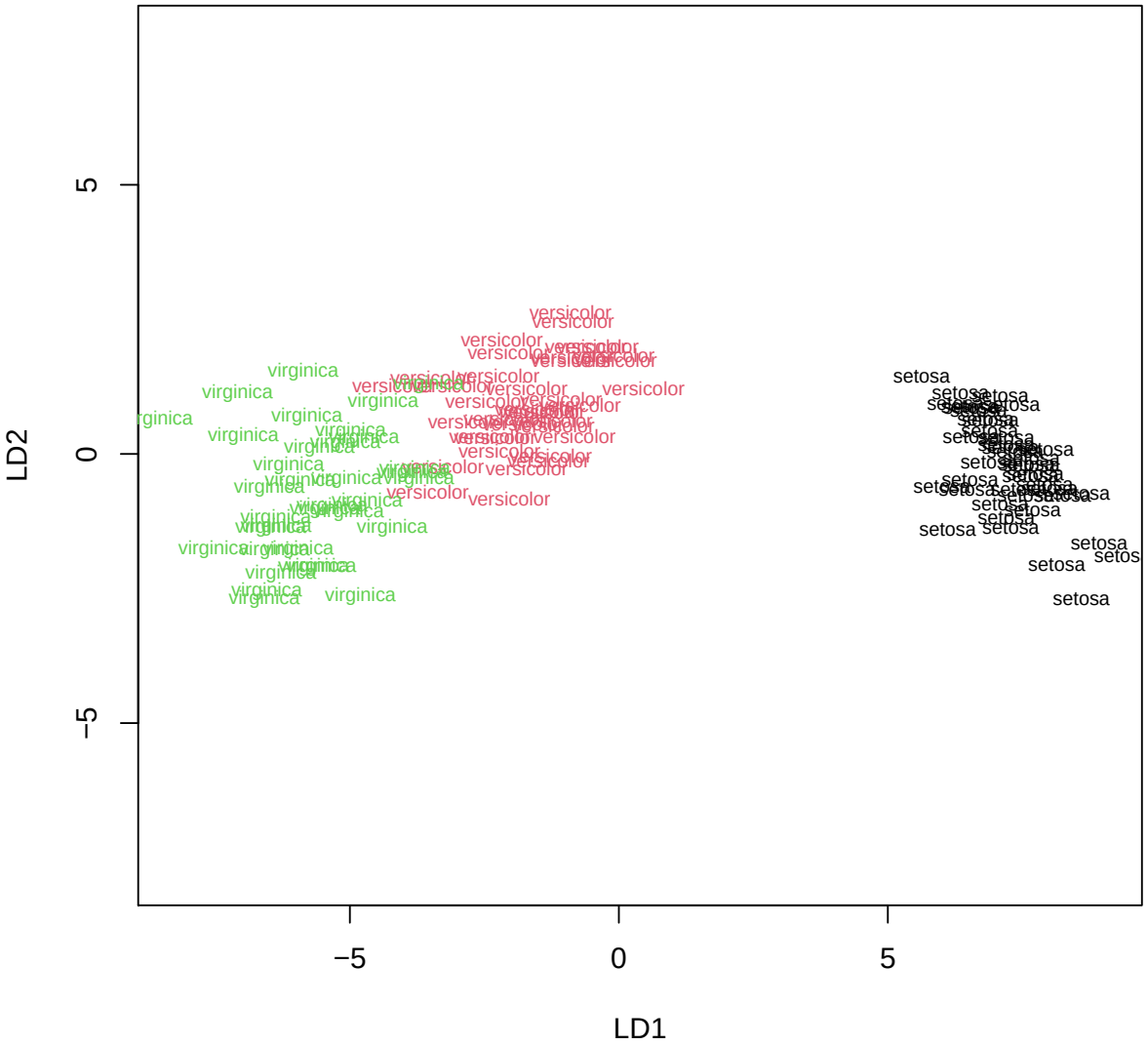
Por lo tanto las funciones discriminantes están dadas por

$$0.86SL + 1.25SW - 2.13PL - 2.70PW$$ **coordenada X**

$$-0.70SL - 1.55SW + 1.70PL - 3.81PW$$ **coordenada Y**

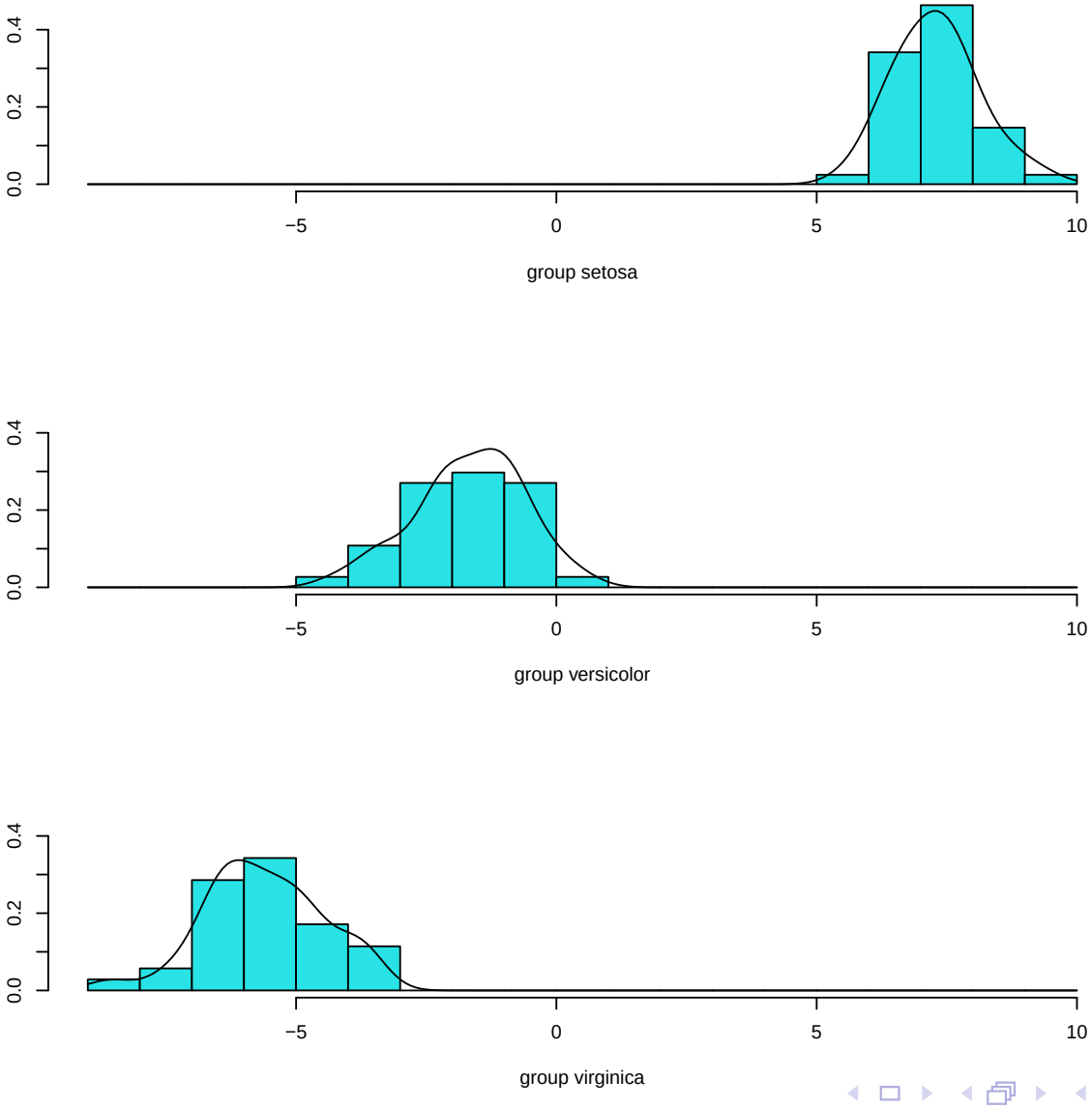
Análisis Discriminante lineal

```
plot(lda.iris,col=as.integer(train$Species))
```



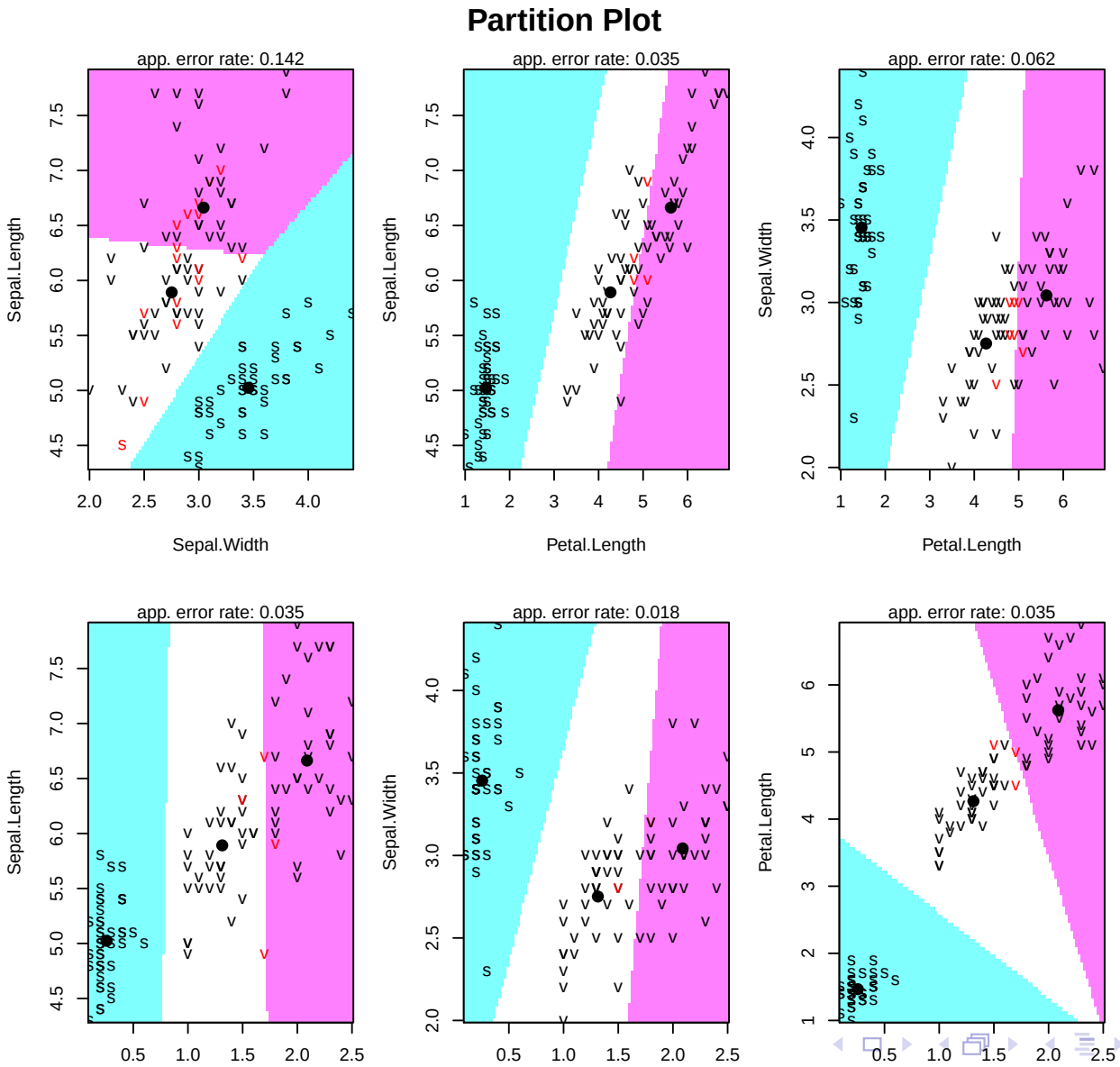
Análisis Discriminante lineal

```
plot(lda.iris, dimen = 1, type = "b")
```



Análisis Discriminante lineal

```
partimat(Species ~ Sepal.Length + Sepal.Width +
Petal.Length + Petal.Width, data=train, method="lda")
```



Análisis Discriminante lineal

Falta realizar la clasificación

```
lda.test = predict(lda.iris),test
test$lda = lda.test$class
table(test$lda,test$Species)
```

ETIQUETAS REALES

ETIQUETAS PREDICHAS	setosa	versicolor	virginica
setosa	9	0	0
versicolor	0	13	2
virginica	0	0	13

matriz de confusion

Hay dos observaciones virginica clasificadas como versicolor.

Función Discriminante Cuadrática

Si las matrices Σ_k con $k = 1, \ldots, K$ son distintas, entonces queda una ecuación cuadrática, no lineal. Sea

$$\delta_k(\mathbf{x_0}) = -\frac{1}{2}\log\left(|\Sigma_k|\right) - \frac{1}{2}(\mathbf{x_0} - \mu_k)'\Sigma_k^{-1}(\mathbf{x_0} - \mu_k) + \log(\pi_k).$$

La frontera de decisión entre las clases k y l está dada por $\{\mathbf{x} : \delta_k(\mathbf{x}) = \delta_l(\mathbf{x})\}$

- las estimaciones de los parámetros se hacen de manera análoga al caso lineal.
- las matrices de covarianza hay que estimarlas por separado para cada grupo, lo que implica, que la cantidad de datos tiene que ser muy elevada. En el caso lineal la cantidad de parámetros a estimar es de orden p, mientras que en el caso cuadrático es de orden p^2.

Función Discriminante Cuadrática

Para determinar si las matrices de covarianza son iguales, podemos realizar el **Test de Box**

En este caso,

$$H_0 : \Sigma_1 = \cdots = \Sigma_K.$$

Estadístico: Consideramos el estimador *pooled* de la matriz de covarianza bajo H_0,
$S = \frac{1}{n-K}\sum_{i=1}^{K}(n_i - 1)S_i$, donde n_i es el tamaño de la población i y $n = \sum_{i=1}^{n} n_i$. Luego,

$$\begin{aligned} M &= (n-K)\ln(|S|) - \sum_{i=1}^{K}(n_i - 1)\ln(|S_i|) \\ c &= \frac{2p^2 + 3p - 1}{6(p+1)(K-1)}\left(\sum_{i=1}^{K}\frac{1}{n_j - 1} - \frac{1}{n-K}\right) \end{aligned}$$

p indica la dimensión del espacio.

Función Discriminante Cuadrática

Luego,

$$M(c-1) \sim \chi^2_{df},$$

donde $df = (p(p+1)(m-1))/2$.
Rechazamos H_0 si

$$M(c-1) > \chi^2_{df,1-\alpha}.$$

Función Discriminante Cuadrática

Continuamos con el ejemplo de los lirios. En primer lugar deberíamos testear si hay homogeneidad en las matrices de covarianza son distintas. Luego usamos el test M de Box, se encuentra en la librería `rstastix` de R

```
box_m(iris[, -5], iris[, 5])
```

H0: Todos los grupos tienen la misma matriz de covarianza

	statistic	p.value	parameter	method
1	140.94	0.00	20.00	Box's M-test for Homogeneity of Co

Rechazo el supuesto de homogeneidad de matrices, deberíamos hacer la clasificación usando un discriminante cuadrático.

Función Discriminante Cuadrática

cuadratic discriminational analysis

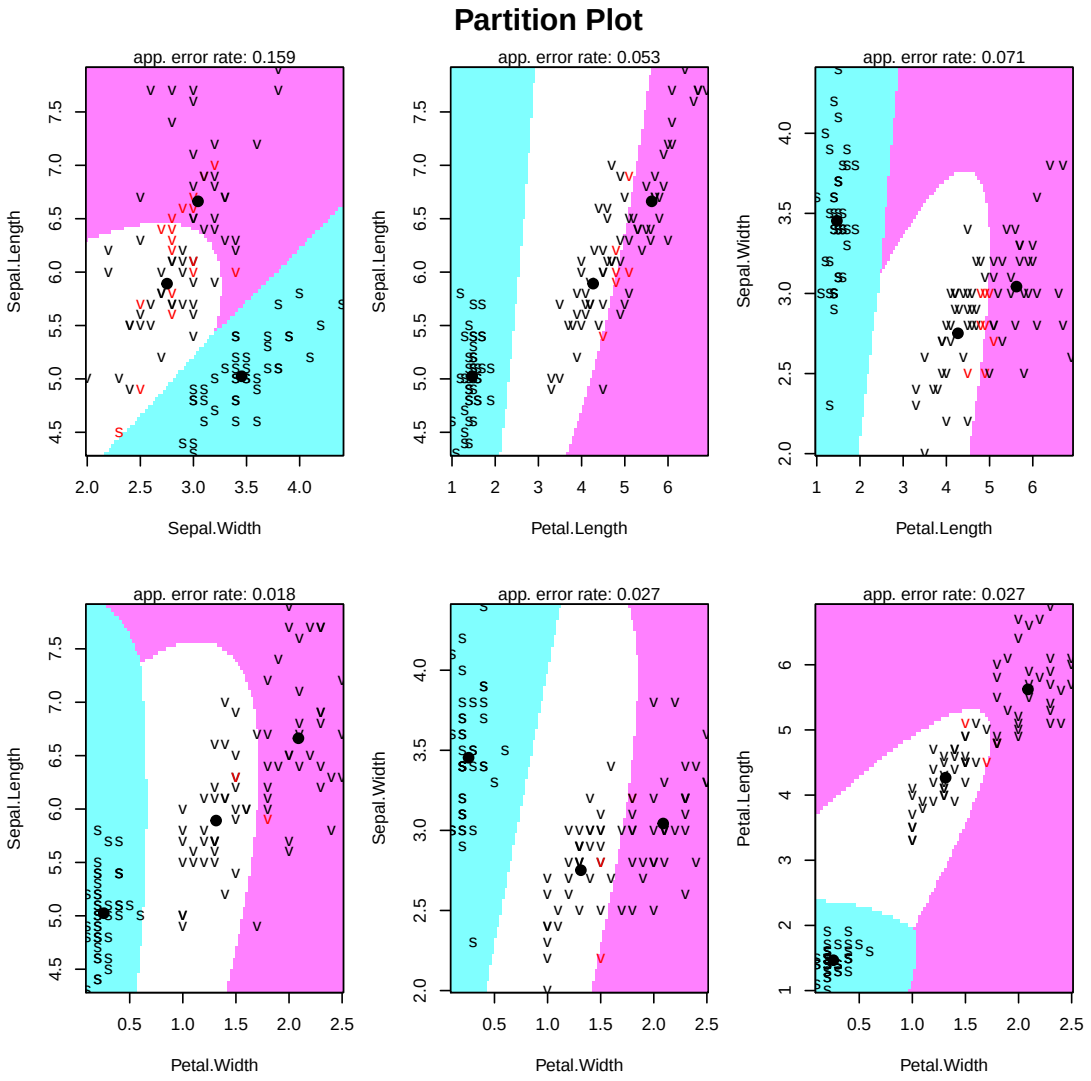
```
qda.iris = qda(Species ~ Sepal.Length + Sepal.Width
+ Petal.Length + Petal.Width, train)
qda.test =predict(qda.iris,test)
test$qda = qda.test$class
table(test$qda,test$Species)
```

	setosa	versicolor	virginica
setosa	9	0	0
versicolor	0	13	0
virginica	0	0	15

No hay observaciones mal clasificadas.

Función Discriminante Cuadrática

```
partimat(Species ~ Sepal.Length + Sepal.Width +
Petal.Length + Petal.Width, data=train, method="qda")
```



Funciones Discriminantes

- Presentan buen comportamiento en problemas con grandes bases de datos.
- Las estimaciones gaussianas son muy estables.
- En cuanto al compromiso sesgo-varianza.
  - LDA, presenta alto sesgo, pero baja varianza, importante varianzas iguales.
  - QDA, presenta más varianza, ya que hay que estimar muchos parámetros. Aunque menos que en alternativas no paramétricas.

Este approach, baja un poco el sesgo, respecto al metodo lineal, pero necesito una mayor cantidad de datos para utilizarlo bien, de lo contrario, voy a tener una alta varianza de los resultados, siendo no consistente el clasificador

Naive Bayes

Si la dimensión del espacio es alta, en comparación con el número de observaciones por clase, la estimación de la densidad resulta una tarea difícil. En estos casos, para resolver el problema en forma viable se suelen hacer supuestos que sobre la estructura de los datos.

En este caso, se asume que para cada clase k las variableas de X_k son independientes, es decir, que se pueden expresar como producto de las distribuciones marginales.

$$f_k(\mathbf{x}) = \Pi_{i=1}^{n} f_{ik}(x_i).$$

Las densidades marginales univariadas se pueden estimar:

- Asumiendo alguna distribución paramétrica y ajustando los parámetros.
- Mediante una distribución no paramétrica, por ejemplo, usando Kernels.

Naive Bayes

Si bien las distribuciones marginales se estiman con alto sesgo, pero la varianza es baja, esto equilibra y hace que en muchos casos la performance sea buena en comparación con métodos más sofisticados.

Naive Bayes

Retomamos el ejemplo de iris data

```
nb.iris = naiveBayes(Species ~ Sepal.Length +
Sepal.Width + Petal.Length + Petal.Width, train)
nbiris.test = predict(nb.iris,test)
table(nbiris.test,test$Species)
xtable(table(nbiris.test,test$Species))
```

	setosa	versicolor	virginica
setosa	9	0	0
versicolor	0	13	2
virginica	0	0	13

Hay dos observaciones virginica clasificadas como versicolor.

Observación: Los métodos que vimos hasta el momento proponen clasificar una nueva observación en el grupo cuya probabilidad a priori sea más alta.
Supongamos que tenemos dos grupos A y B entonces clasificamos a una observación $\mathbf{x}_0$ en el grupo en el grupo A si

$$P(Y = A|\mathbf{x} = \mathbf{x}_0) > 0.5.$$

Si queremos privilegiar alguno de los dos errores, porque consideramos que es más grave, podemos mover el corte. Por ejemplo ser más restrictivos a la hora de clasificar a una observación $\mathbf{x}_0$ en el grupo A y por ejemplo clasificar a $\mathbf{x}_0$ en el grupo en el grupo A si

$$P(Y = A|\mathbf{x} = \mathbf{x}_0) > 0.8.$$

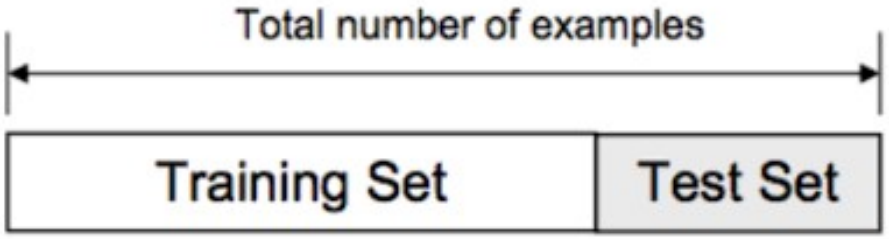
.

Medidas de Performance

Cómo decimos con que regla de clasificación nos quedamos? Dicha decisión debe basarse en el conocimiento específico del problema, cómo por ejemplo la información sobre los costos asociados con el incumplimiento de pagos.

Muestras de entrenamiento y validación

Idea ingenua: medir la eficacia (accuracy), el porcentaje de datos clasificados correctamente, en la muestra de entrenamiento.
De este modo se **sobreestima** el desempeño de la regla de aprendizaje.
Separar al azar la muestra dos, una parte (mayoritaría) se usa para entrenar al algoritmo y la restante para medir su eficacia.



Muestras de entrenamiento y validación

El error en la muestra de entrenamiento y de validación es distinto, típicamente es mayor en la muestra de validación.

k-Fold Cross Validation

- Separar al azar los datos en k folds del mismo tamaño.
- Para $i = 1, \ldots, k$ entrenar sobre todos los folds menos el i-ésimo. Usar el i-ésimo para validar, medir la **eficacia**.

k-Fold Cross Validation

Para comparar dos procedimientos.
Usar la misma partición de k folds para los dos procedimientos

- Se pueden comparar las accuracies medias. **"Calcular el promedio"**
- Se pueden utilizar criterios que comparen las medias menos un desvío estándar.
- Se pueden hacer test de rangos signados para muestras apareadas, entre los vectores de las accuracies para cada uno de los procedimientos comparados. Ventaja: detecta si la diferencia es significativa.

Los objetivos que podemos tener son dos:

- Seleccionar entre varios modelos aquel que tenga mejor ajuste (R^2, accuracy, etc)
- Habiendo elegido un modelo, estimar su error de predicción en un nuevo conjunto de datos.

k-Fold Cross Validation

Los objetivos que podemos tener son dos:

- Seleccionar entre varios modelos aquel que tenga mejor ajuste (R^2, accuracy, etc)
- Habiendo elegido un modelo, estimar su error de predicción en un nuevo conjunto de datos.

Un problema que puede surgir es que estemos sobreestimado el desempeño de un modelo, esto será evidente cuando lo utilicemos en un nuevo conjunto de datos.

k-Fold Cross Validation

Si tenemos cantidad suficiente de datos, dividir a la muestra en tres (típicamente 50%/25%/25%)

Ajustamos los modelos con las muestras de entrenamiento y validación. Una vez elegido el modelo analizamos su desempeño con la muestra de prueba. Esto dará una visión más realista del desempeño.

k-Fold Cross Validation

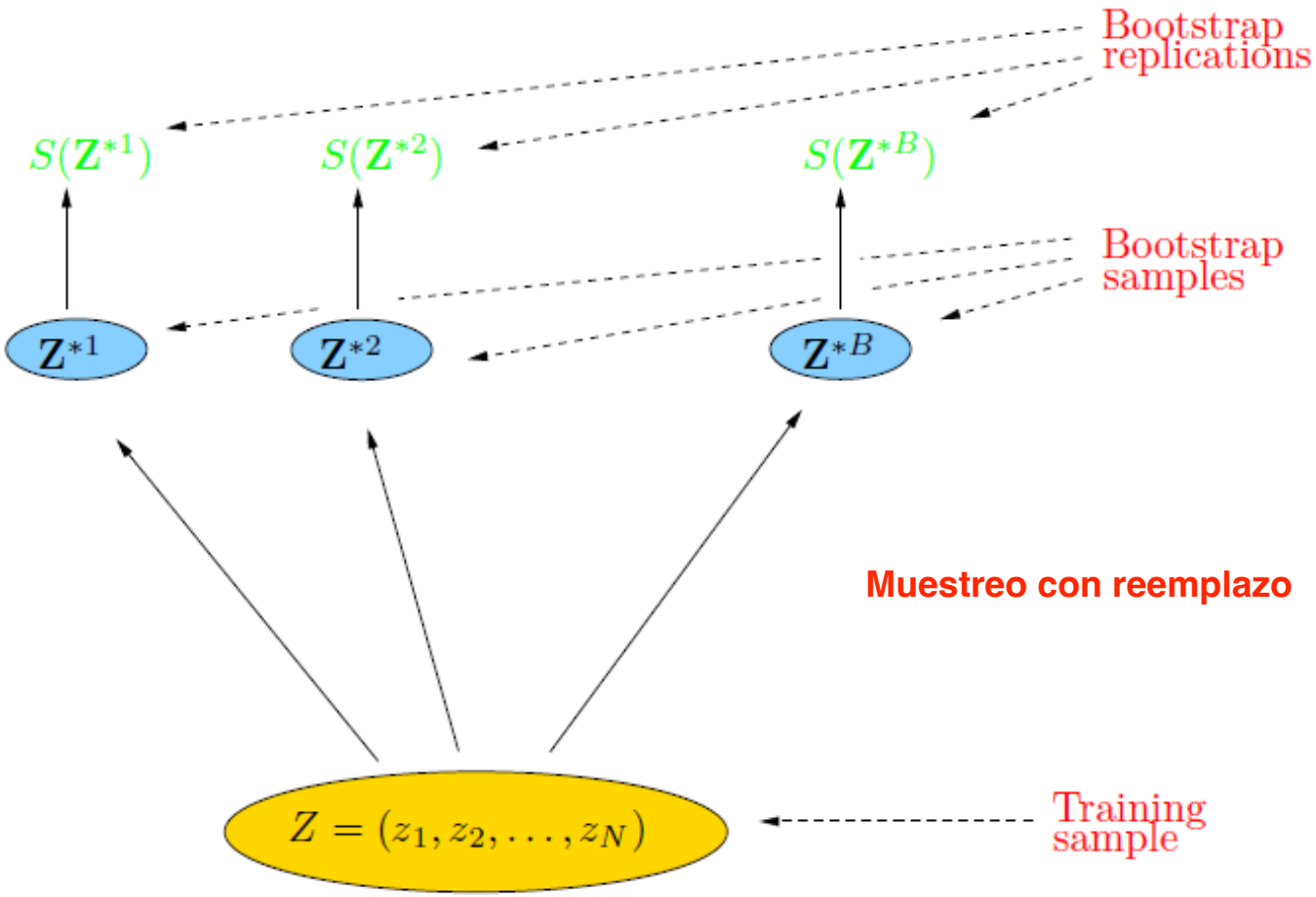
Importante: para que CV sea válido hay que aplicarlo antes de realizar cualquier instancia de selección de variables o filtrado. Si se hacen procedimientos de selección de variables antes de CV deben ser procedimientos que no tengan en cuenta las etiquetas (no supervisados).

Bootstrap

Es otra estrategia para estimar el error esperado de predicción. Idea general: Tenemos una muestra $Z = \{z_1, \ldots, z_n\}$, donde $z_i = (x_i, y_i)$. Extraemos B muestras con reposición. $Z_1^*, \ldots, Z_B^*$. A partir de las muestras bootstrap podemos estimar cualquier estadístico a partir de $Z_1^*, \ldots, Z_B^*$, por ejemplo el R^2, accuracy.

En muchas ocaciones este procedimiento produce sobreajuste ya si se usa la misma muestra para ajuste y validación. Se recomienda considerar muestras disjuntas para ambas etapas, utilizar en la etapa de evaluación aquellas observaciones que no fueron utilizadas en la muestra bootstrap.

Bootstrap


Bootstrap replications
$S(\mathbf{Z}^{*1})$
$S(\mathbf{Z}^{*2})$
$S(\mathbf{Z}^{*B})$
Bootstrap samples
$\mathbf{Z}^{*1}$
$\mathbf{Z}^{*2}$
$\mathbf{Z}^{*B}$
Muestreo con reemplazo
$Z = (z_1, z_2, \ldots, z_N)$
Training sample

Bootstrap

Típicamente se utiliza en un amplia gama de técnicas de clasificación para **estimar la variabilidad** que de otro modo es difícil de obtener.

Ejemplo: queremos invertir dinero en dos activos X e Y de modo tal de tener el máximo retorno. Invertiremos una fracción de nuestro dinero α en X y la restante en Y. La inversión en cada activo tiene una variabilidad. Luego, queremos minimizar

$$var\left(\alpha X + (1-\alpha)Y\right) = \alpha^2\sigma_X^2 + (1-\alpha)^2\sigma_Y^2 + 2\alpha(1-\alpha)\sigma_{XY}.$$

Luego, el riesgo se minimiza si,

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

Estimando las varianzas y covarianzas y via plug-in tenemos un estimador de α

Bootstrap

Queremos estimar la varianza de α.
Tomamos $B = 1000$ muestras bootstrap de la muestra original.
Luego tenemos $\widehat{\alpha_1^*}, \ldots, \widehat{\alpha_{1000}^*}$
Luego podemos estimar

$$\bar{\alpha} = \frac{1}{B} \sum_{i=1}^{B} \widehat{\alpha_i^*}$$

y la variablidad la estimamos

$$\sqrt{\frac{1}{B-1} \sum_{i=1}^{B} \left(\widehat{\alpha_i^*} - \bar{\alpha} \right)^2}$$

Medidas de Performance

Accuracy: no distingue entre los errores que cometemos, no brinda información.

Total · Argentina

Confirmados	Personas recuperadas	Muertes
167 k +4890	72,575	3059 +120

Ubicación	Confirmados	Personas recuperadas	Muertes
Provincia de Buenos Aires	98,228 +3351	47,805	1592 +59
Ciudad de Buenos Aires	78,935	22,893	1429
Provincia del Chaco	3358 +32	2501	139 +2
Provincia de Córdoba	1866 +99	495	44 +0
Jujuy	1764 +136	254	47 +1

El signo "+" indica los casos nuevos que se reportaron ayer · Actualizadas hace 5 horas · Fuente: Wikipedia · Información sobre estos datos

La tasa de muerte es de 1.8, máximo error global (todos se salvan) de classificación es 1.8

Medidas de Performance

Matriz de Confusión Consideremos un problema de clasificación con dos categorías. Se hace un test para determinar si un individuo tiene determinada enfermedad.

REALES / OBSERVADOS

PREDICHOS

	Enfermo	Sano
Pred Enfermo	Verdadero Positivo, tp	Falso Positivo, fp
Pred Sano	Falso Negativo, fn	Verdadero Negativo, tn

Los dos errores no son igualmente graves, en general se quiere privilegiar tener pocos errores de los graves. En el caso del COVID serían los fn, se predice como sano pero tiene COVID, no se trata y sigue contagiando.

Medidas de Performance

Especificidad

También se llama true negative rate Mide la proporción de **positivos** reales que están bien clasificados. El porcentaje de personas sanas identificadas como que no tienen la enfermedad.

$$Especifidad = \frac{tn}{tn + fp}$$

Si el resultado es negativo cuál es la probabilidad de que realmente no tenga la enfermedad.

La **tasa de falso positivo** se define como **1-especifidad.**

Si esta sano, cual es la probabilidad de que realmente no detecte la enfermedad

Medidas de Performance

Sensitividad

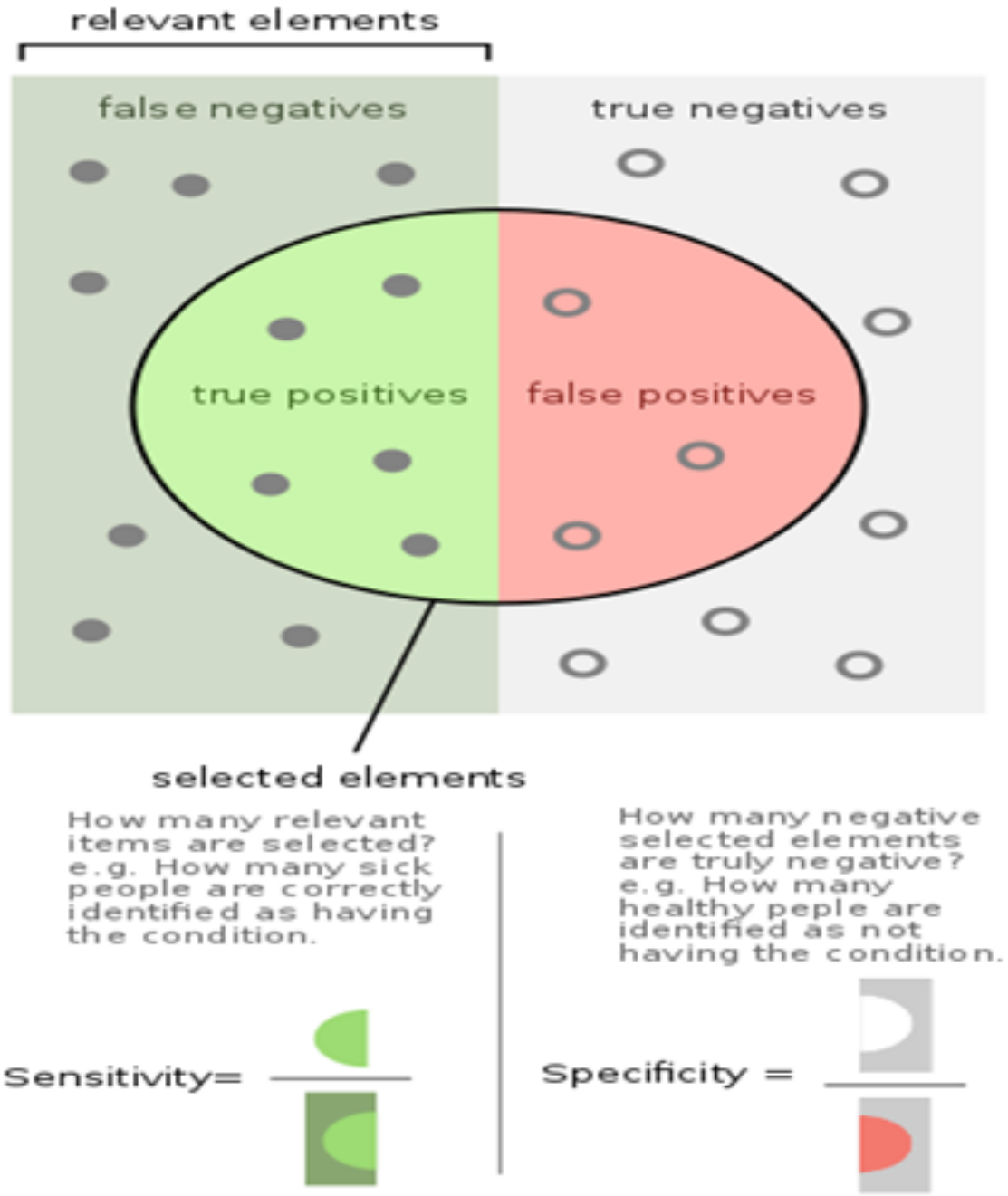
También se conoce como recall o tasa de verdadero positivo.
Mide la proporción de **negativos** reales que están bien clasificados.
El porcentaje de personas enfermas que están identificadas de ese modo.

$$Sensitividad = \frac{tp}{tp + fn}$$

Si el resultado es positivo cuál es la probabilidad de que realmente tenga la enfermedad.

Si tiene la enfermedad, cual es la probabilidad de que el resultado sea positivo

Medidas de Performance



Medidas de Performance

Otras medidas:

- $$\text{Precisión} = \frac{tp}{tp + fp}$$
- $$F_1 - score = 2\frac{\text{Precisión } \times \text{ Recall}}{\text{Precisión } + \text{ Recall}}$$
- En general

$$F_\beta = (1 + \beta^2)\frac{\text{Precisión } \times \text{ Recall}}{(\beta^2 \text{ Precisión }) + \text{ Recall}},$$

$\beta > 1$ da más importancia a la Precisión, mientras que $\beta < 1$ al Recall.

Medidas de Performance

La **Matriz de Confusión** puede extenderse al caso en que la clasificación sea entre varias categorías.
Se quiere diferenciar entre una **P**, una **B** y una **D**.

	P	**B**	**D**
Pred **P**	**PP**	**BP**	**DP**
Pred **B**	**PB**	**BB**	**DB**
Pred **D**	**PD**	**BD**	**DD**

Para cada categoría puedo dar las definiciones marginales de los estadísticos que definimos (Especificidad, Sensitividad, Precisión, etc) de cada clase vs las demás. A modo de ejemplo, Sensitividad de la letra **P**

$$\text{Sensitividad} = \frac{\mathbf{PP}}{\mathbf{PP} + \mathbf{PB} + \mathbf{PD}}$$

Medidas de Performance

Retomamos el ejemplo de los datos **iris**, nos quedamos con la última clasificación que hicimos que fue usando **naive bayes.** Primero armo la matriz de confusión, prestar atención que tengo que poner en las filas los datos de la muestra y en las columnas los predichos.

```
tab=table(test$pecies, nbiris.test, dnn = c("Real",
"Predicha"))
library(caret)
cm.nb.iris=confusionMatrix(tab)
cm.nb.iris$table
```

	setosa	versicolor	virginica
setosa	9	0	0
versicolor	0	13	0
virginica	0	2	13

Medidas de Performance

```
cm.nb.iris$overall
```

Overall Statistics

- Accuracy : 0.9459
- 95% CI : (0.8181, 0.9934)
- No Information Rate : 0.4054
- P-Value [Acc > NIR] : $4.618e-12$
- Kappa : 0.9176
- Mcnemar's Test P-Value : NA

Medidas de Performance

```
cm.nb.iris$byClass
```

	Sensitivity	Specificity	PosPredValue	NegPredValue	Precision	Recall	F1
setosa	1.00	1.00	1.00	1.00	1.00	1.00	1.00
versicolor	0.87	1.00	1.00	0.92	1.00	0.87	0.93
virginica	1.00	0.92	0.87	1.00	0.87	1.00	0.93

Medidas de Performance

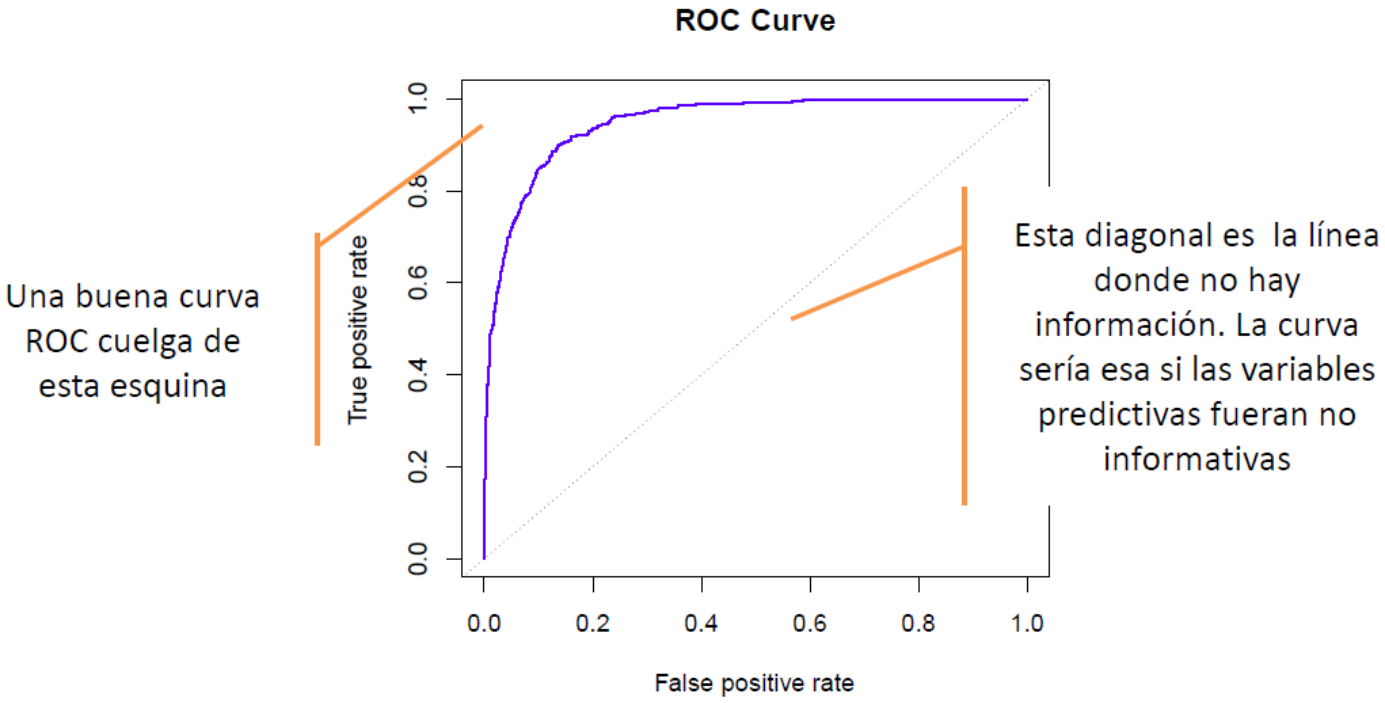
```
cm.nb.iris$byClass
```

	Prevalence	DetectionRate	DetectionPrevalence	BalancedAccuracy
setosa	0.24	0.24	0.24	1.00
versicolor	0.41	0.35	0.35	0.93
virginica	0.35	0.35	0.41	0.96

Curva ROC

- La curva ROC (receiver operating characteristics) es un gráfico popular para mostrar simultáneamente los dos tipos de errores para todos los umbrales posibles.
- La curva ROC resume el rendimiento general de un clasificador como el analizando el área debajo de la curva.
- Las curvas ROC son útiles para comparar diferentes clasificadores, ya que tienen en cuenta todos los umbrales posibles.

Curva ROC



A partir de este gráfico también se puede definir **AUC**, que es el área bajo la curva. Es un número entre 0 y 1, si las variables fueran no informativas, entonces sería 0.5.

Curva ROC

Consideramos el conjunto de datos **BreastCancer** de la librería **mlbench**. Contiene datos de 699 sobre tumores en la mamas y se quieren clasificar como *benignos* o *malignos*
Contamos con 9 variables, que son nominales u ordinales.

- Id Sample code number
- Cl.thickness Clump Thickness
- Cell.size Uniformity of Cell Size
- Cell.shape Uniformity of Cell Shape
- Marg.adhesion Marginal Adhesion
- Epith.c.size Single Epithelial Cell Size
- Bare.nuclei Bare Nuclei
- Bl.cromatin Bland Chromatin
- Normal.nucleoli Normal Nucleoli
- Mitoses Mitoses
- Class Class

Curva ROC

Aplicamos un clasificador lineal.

```
library(mlbench)
data("BreastCancer")
bc=na.omit(BreastCancer[,-1])
training_sample = sample(c(TRUE, FALSE), nrow(bc),
replace = T, prob = c(0.75,0.25))
train = bc[training_sample, ]
test = bc[!training_sample, ]
lda.bc = lda(Class ~ ., train)
lda.test.bc = predict(lda.bc,test)
```

Curva ROC

```
confusion = table(test$Class, lda.test.bc$class, dnn
= c("Real", "Predicha"))
```

	benign pred	malignant pred
benign	113	4
malignant	4	58

Curva ROC

- Accuracy : 0.9553
- 95% CI : (0.9138, 0.9805)
- No Information Rate : 0.6536
- P-Value [Acc > NIR] : $< 2e - 16$
- Kappa : 0.9013
- Mcnemar's Test P-Value : 1
- Sensitivity : 0.9658
- Specificity : 0.9355
- Pos Pred Value : 0.9658
- Neg Pred Value : 0.9355
- Prevalence : 0.6536
- Detection Rate : 0.6313
- Detection Prevalence : 0.6536
- Balanced Accuracy : 0.9506
- 'Positive' Class : benign

Curva ROC

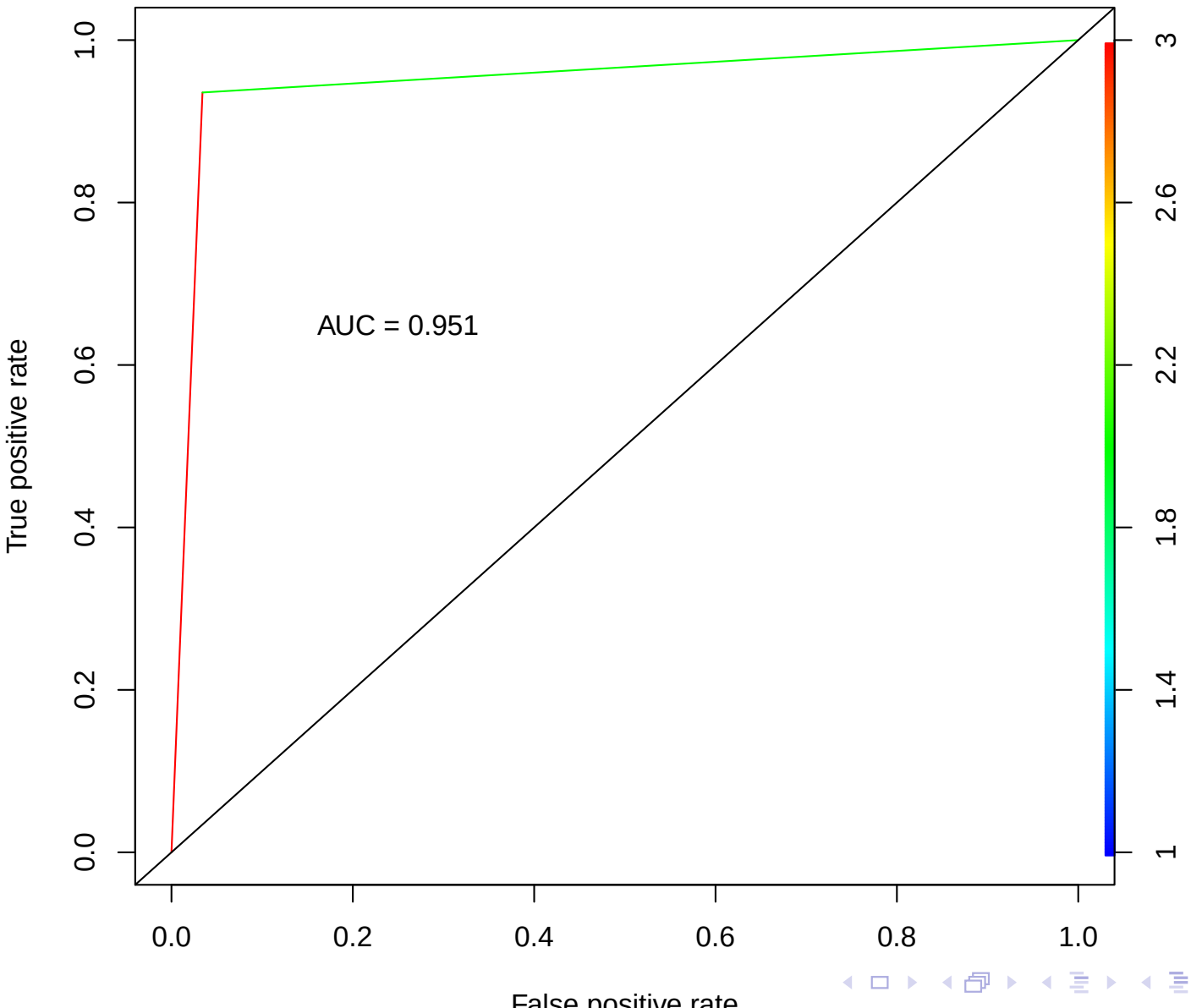
Armamos la curva ROC

```
library(ROCR)
df = data.frame(predictions=
as.numeric(lda.test.bc$class),labels=as.numeric(test$Class)
```

Hay que tener cuidado porque no funciona si se pone como factores

```
pred = prediction(as.numeric(df$predictions),
df$labels)
perf = performance(pred,"tpr","fpr")
auc.pred = performance(pred, measure = "auc")
auc.pred = auc.pred@y.values
plot(perf,colorize=TRUE)
abline(a=0,b=1)
text(x = .25, y = .65 ,paste("AUC = ",
round(auc.pred[[1]],3), sep = ""))
```

Curva ROC


AUC = 0.951
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
1
1.4
1.8
2.2
2.6
3

M. Kuhn, k. Johnson.
Applied Predicting Modelling.
Springer-Verlag, 2015.
Capítulo 11 y 12.

F. Harrell.
Regression Modelling Strategies.
Springer-Verlag, 2015.
Capítulo 5.

T. Hastie, R. Tibshirani, J. Friedman.
Elements of Statistical Learning, 2nd Ed.
Springer-Verlag, 2009.
Capítulo 4 y 7.

J. Garret, D. Witten, T. Hastie, R. Tibshirani.
An Introduction to Statistical Learning.
Springer-Verlag, 2013.
Capítulo 4 y 5.