Elements of Machine Learning

Clasificación

MSc. Diego Porres

UFM | UNIVERSIDAD FRANCISCO MARROQUÍN

Febrero 2019

Some of the figures in this presentation are taken from *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Some of the figures in this presentation are taken from *The Elements of Statistical Learning* (Springer, 2009) with permission from the authors: T. Hastie, R. Tibshirani, and J. Friedman

Intro (1)

- Pasamos ahora al caso en donde la respuesta Y es cualitativa, por lo que la denotaremos por G.
- A las variables cualitativas también se les llama *categóricas*.
- **Objetivo:** *clasificar* a la observación X en una de las K distintas categorías (no ordenadas).
- G tomará valores en un conjunto finito $\mathcal{G}$, por lo que:

$$G : \mathbb{R}^p \to \mathcal{G} = \{1, \ldots, K\}$$

  - En general estamos más interesados en calcular las probabilidades que X pertenezca a cada clase.

Intro (2)

- A las técnicas usadas para clasificar datos se les conoce como *clasificadores*.
- Analizaremos a los siguientes clasificadores:
  - Regresión logística
  - Análisis discriminante lineal (LDA)
  - K-vecinos cercanos (KNN)
- Veremos otros clasificadores más adelante.

Intro (3)

- Los problemas de clasificación son más comunes que los de regresión.
- Algunos ejemplos son:
  - Si una persona ingresa a la sala de emergencia de un hospital con un conjunto de síntomas debido a uno de tres posibles enfermedades, ¿qué enfermedad es la que padece?
  - El servicio de banca en línea de un banco determina si una transacción es o no fraudulenta, tomando a la dirección IP, historial de transacciones, hora de la transacción, etc.
  - Clasificar a un correo como normal o *spam*, basándonos en las palabras en el cuerpo del correo y del título, etc.
  - Predecir si un individuo va a incurrir en un `impago` (`default`) en su tarjeta de crédito, dado su salario anual, balance mensual en su tarjeta de crédito, etc.

Datos de Default

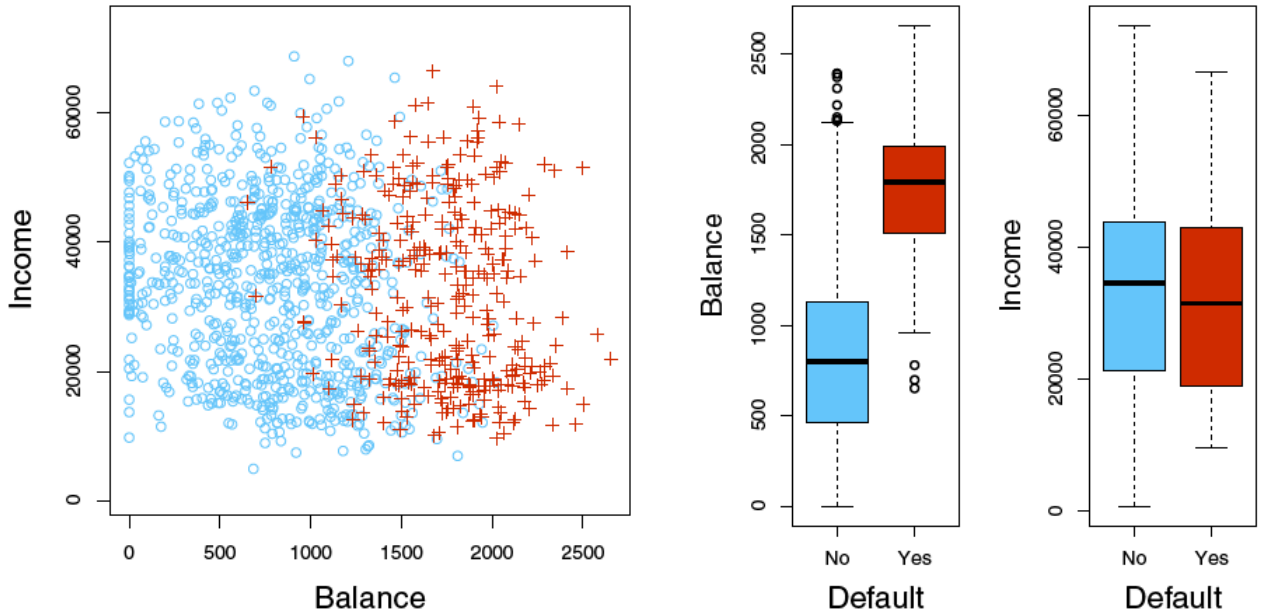

Figura 1: Graficamos los datos de Default (datos simulados). A la izquierda, vemos la relación entre el balance en la tarjeta de crédito vs. el ingreso anual (en \$). Los individuos que incurrieron en un impago se muestran en rojo, mientras los que no se muestran en azul. A la derecha, tenemos diagramas de cajas del balance e ingresos en función del estado de impago. (La división de los datos casi nunca es así de clara en la vida real.)

Intro (4)

- G dividirá a nuestro espacio en regiones etiquetadas de acorde a su clasificación.
- Las fronteras entre estas regiones se les llama *fronteras de decisión*.
- Cuando las fronteras son lineales, denominamos al clasificador como *lineal*.
- Podemos expandir a nuestro conjunto de variables $X_1, \ldots, X_p$ a que incluya también los productos y cuadrados: $X_1^2, \ldots, X_p^2, X_1X_2, X_1X_3, \ldots$.
- Ésto nos agregará $\binom{p}{2} = p(p+1)/2$ variables.
- Fronteras de decisión lineales en este espacio aumentado corresponderán a fronteras de decisión *cuadráticas* en el espacio original.

Fronteras de Decisión

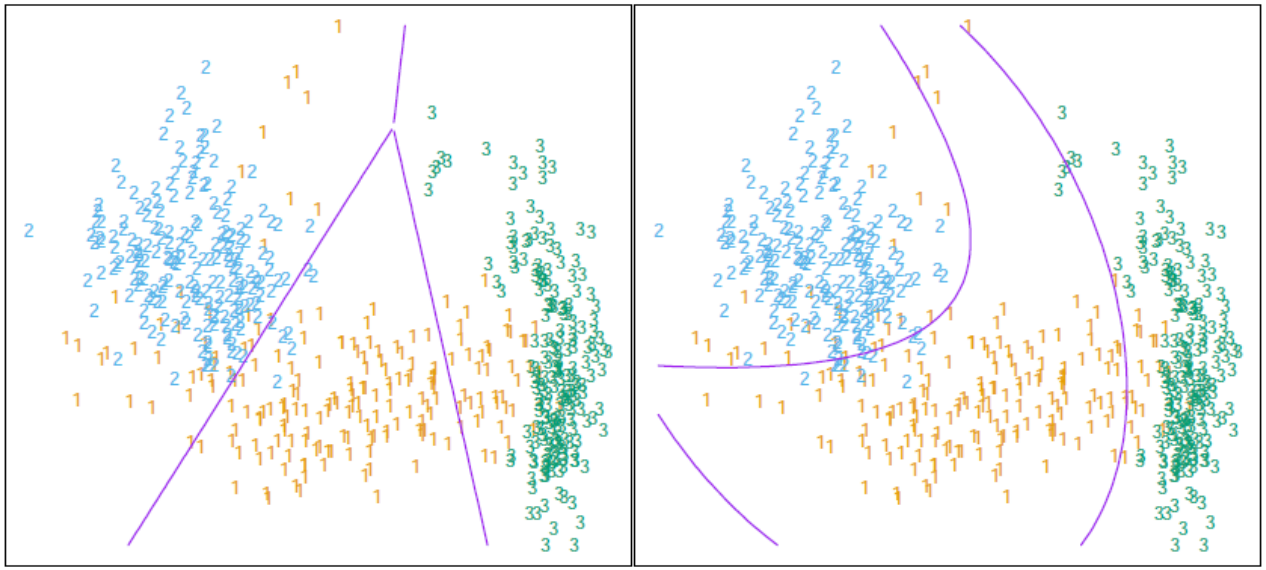

Figura 2: A la izquierda graficamos los datos de tres clases con fronteras de decisión lineales. A la derecha graficamos las fronteras de decisión cuadráticas. Éstas fueron obtenidas al encontrar las fronteras de decisión lineales en el espacio pentadimensional $X_1, X_2, X_1X_2, X_1^2, X_2^2$. Líneas en dicho espacio son curvas cuadráticas en el espacio original.

¿Cuándo aparece una frontera de decisión lineal?

- Nuestro objetivo, entonces, es de aprender una *función discriminante* $\delta_k(x)$ para cada clase k y establecer:

$$G(x) = \arg\max_k \delta_k(x) \tag{1}$$

- G generará una frontera de decisión lineal si existe alguna transformación monótona g de $\delta_k(x)$ que sea lineal.
- Es decir, g es una función monótona tal que

$$g(\delta_k(x)) = \gamma_{k0} + \gamma_k^\top x$$

Un ejemplo de una frontera de decisión lineal

- Por ejemplo, podemos usar a las probabilidades a posteriori $\mathbb{P}[G = k|X = x]$ como nuestras funciones discriminantes para dos clases:

$$\begin{aligned} \delta_1(x) &= \mathbb{P}[G = 1|X = x] = \frac{\exp\left(\beta_0 + \beta^\top x\right)}{1 + \exp\left(\beta_0 + \beta^\top x\right)} \\ \delta_2(x) &= \mathbb{P}[G = 2|X = x] = \frac{1}{1 + \exp\left(\beta_0 + \beta^\top x\right)} \end{aligned} \tag{2}$$

- Podemos aplicar a la transformación monótona (*logit*) $g(p) = \log\left(p/(1-p)\right)$ y vemos que:

$$\log \frac{\mathbb{P}[G = 1|X = x]}{\mathbb{P}[G = 2|X = x]} = \beta_0 + \beta^\top x \tag{3}$$

¿Por qué no usamos regresión lineal? (1)

- Para una respuesta *binaria* (de dos niveles) como en el caso de los datos de `Default` tendremos:

$$G = \begin{cases} 0 & \text{si la persona } \texttt{No} \text{ incurrió en un impago} \\ 1 & \text{si la persona } \texttt{Si} \text{ incurrió en un impago} \end{cases}$$

- ¿Qué pasa si realizamos una regresión lineal de G sobre X y lo clasificamos como `Si` si $\hat{G} > 0.5$?
- En este caso, la regresión lineal realizará un buen trabajo y será equivalente al *análisis discriminante lineal (LDA)*.
  - El $X\hat{\beta}$ obtenido será un estimador de $\mathbb{P}[\texttt{Si}|X]$.
  - Sin embargo, algunas de nuestras estimaciones estarán fuera del intervalo $[0, 1]$.
  - Para ésta tarea, la *regresión logística* es más adecuada.

Fronteras de Decisión

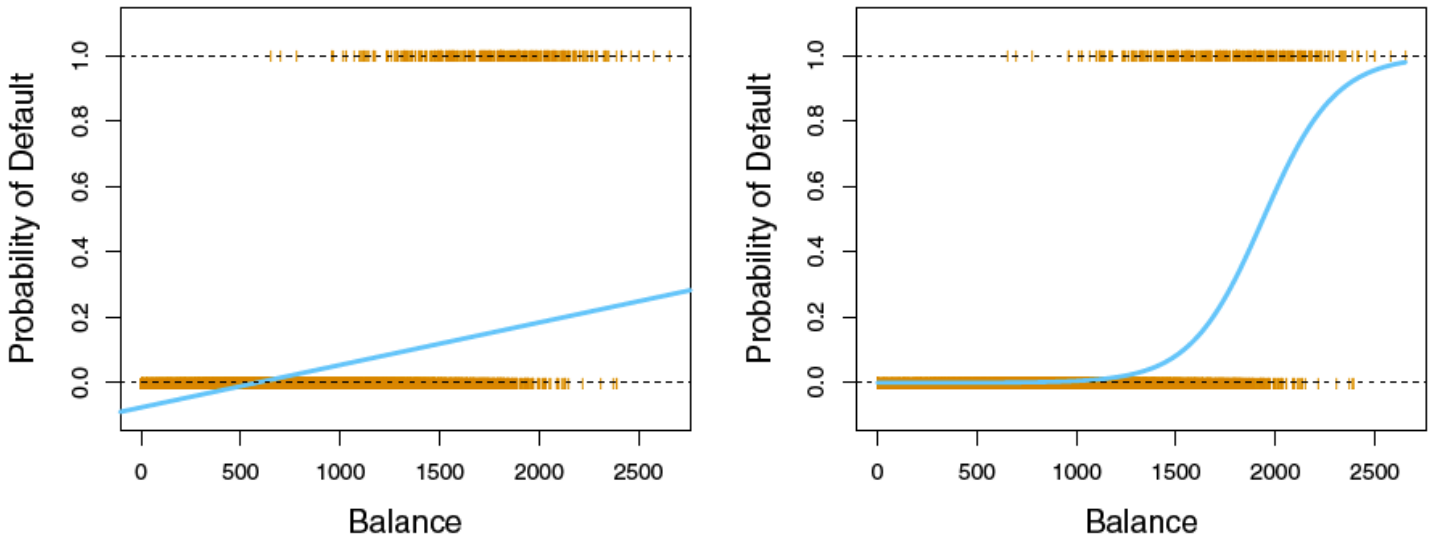

Figura 3: Clasificando a los datos de `Default` en función del `Balance`. A la izquierda, estimamos las probabilidades usando regresión lineal. Nótee que algunas de las probabilidades son *negativas*, mientras que en los datos de `Balance` máximo, la probabilidad llega como máximo al 30%. A la derecha, predecimos la probabilidad de `impago` usando regresión logística. Los datos se presentan de color naranja e indican la verdadera clasificación de los mismos.

¿Por qué no usamos regresión lineal? (2)

- Suponga ahora que tratamos de clasificar, dados los síntomas $X_1, \ldots, X_p$, la condición/enfermedad que sufre un paciente:

$$G = \begin{cases} 1 & \text{si es un } \texttt{infarto} \\ 2 & \text{si es una } \texttt{sobredosis} \\ 3 & \text{si es un } \texttt{ataque epiléptico} \end{cases}$$

- El sólo hecho de haber ordenado a las clases implica que la diferencia entre éstas es la misma.
- El órden utilizado es arbitrario y, de haber usado otro, la regresión lineal predeciría distintos valores en $\mathcal{T}_{\mathsf{Te}}$.
- Por lo tanto, *Regresión logística multinomial* o *Análisis discriminante* son más apropiados.

Fronteras de decisión parar tres clases

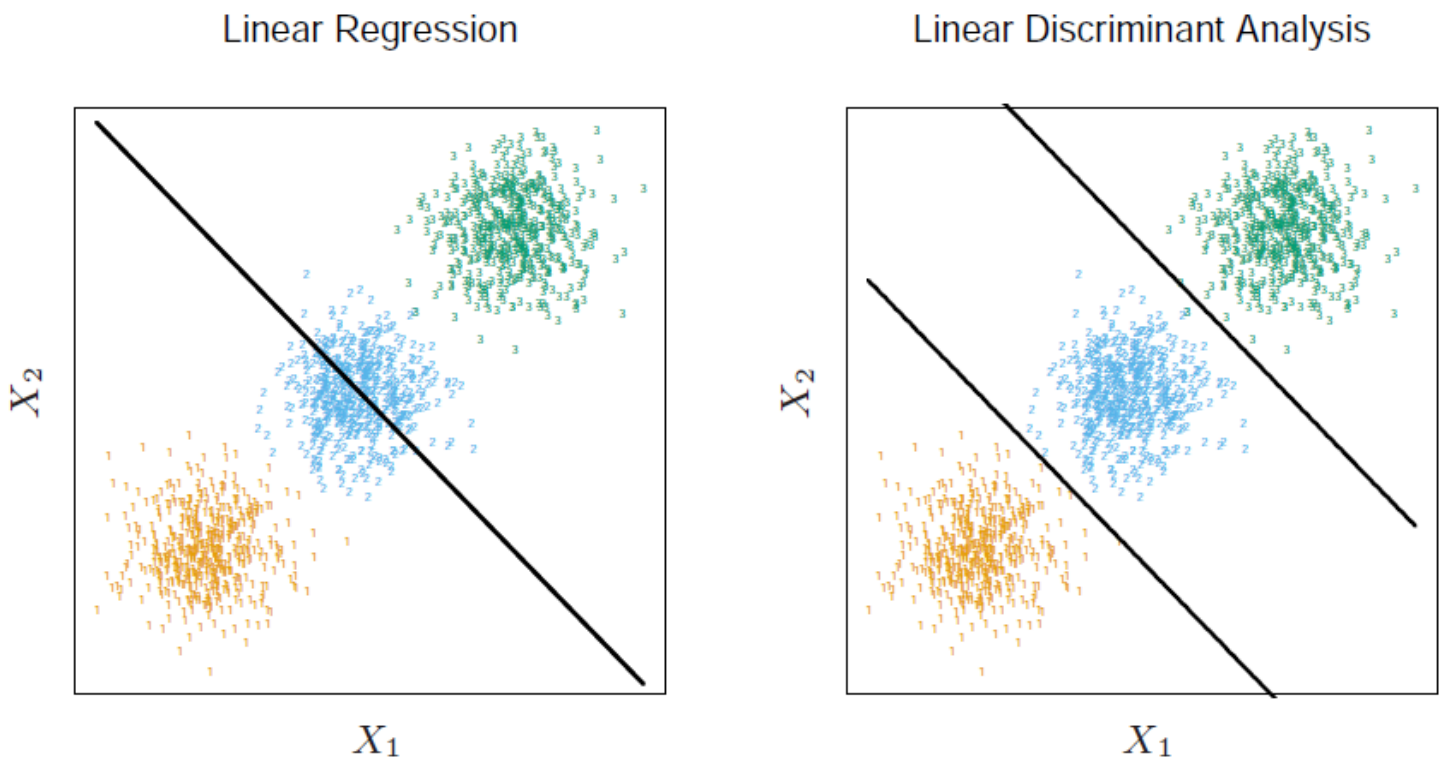

Figura 4: Para datos de tres clases en $\mathbb{R}^2$, a la izquierda vemos que las fronteras de decisión generadas por una regresión lineal *enmascaran* a la clase de enmedio (nunca domina). En cambio, a la derecha vemos las fronteras generadas por el análisis discriminante lineal, el cual separa fácilmente a las clases.

Enmascaramiento de los datos

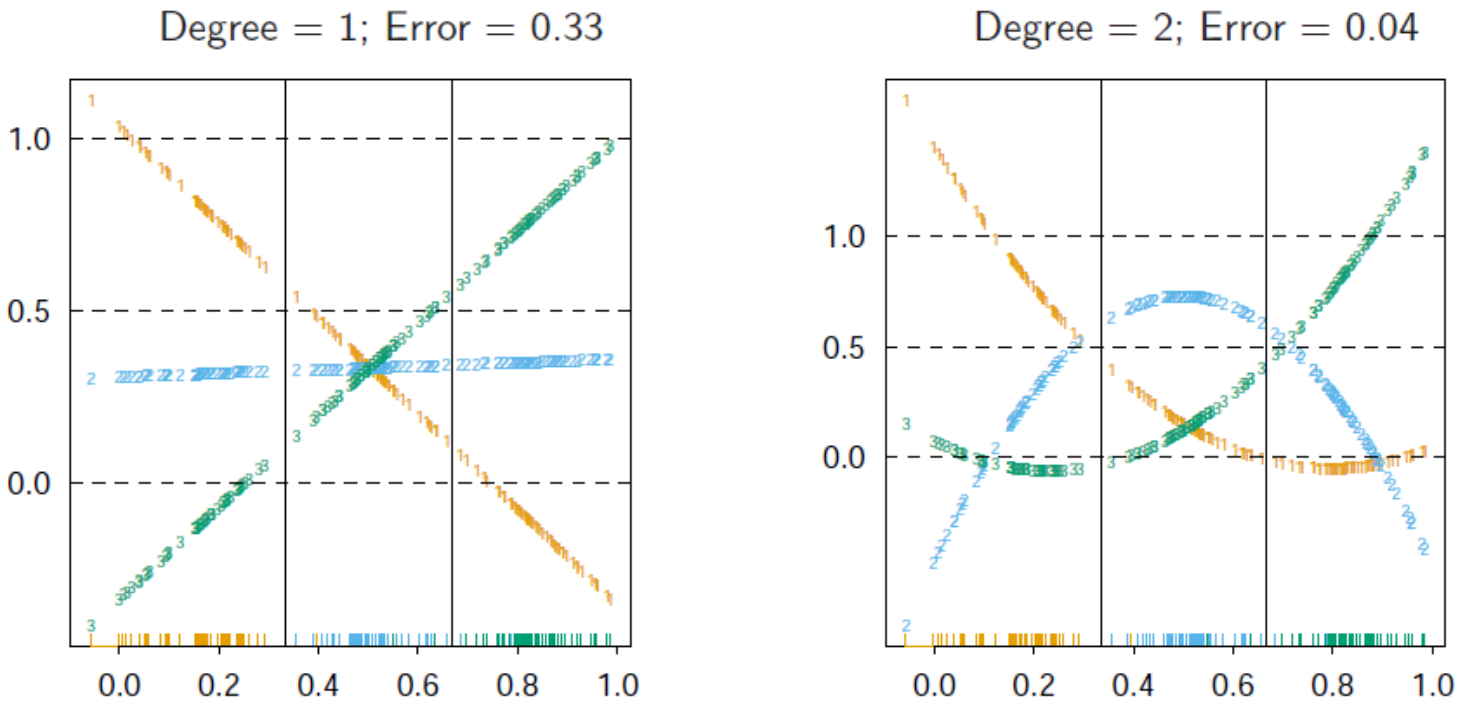

Figura 5: Realizamos *rug plots* para cada clase, y graficamos la probabilidad estimada por la regresión lineal. A la izquierda, vemos que la clase 2 está siempre enmascarada por las otras dos clases. A la derecha, realizamos la regresión lineal con polinomios cuadráticos, lo cual decrece el error de manera considerable. El error de Bayes es de 0.025 y es el realizado por el análisis discriminante lineal.

Regresión Logística

- Consideramos de nuevo a los datos de `Default` cuyas clases definimos como `Si` o `No`.
- No queremos conseguir a G directamente, sino a la probabilidad de que pertenezca a cualquiera de las dos clases, e.g.:

$$p(\texttt{Balance}) = \mathbb{P}[\texttt{Impago} = \texttt{Si} | \texttt{Balance}]$$

- Predecimos que $\texttt{Impago} = \texttt{Si}$ para toda persona que cumpla con $p(\texttt{Balance}) > \alpha$, donde α es un *umbral* previamente establecido.
  - Para algunas aplicaciones basta $\alpha = 0.5$, pero con otras más conservativas, quizá sea mejor $\alpha = 0.1$.

El modelo logístico

- Abreviemos la relación de manera $p(X) = \mathbb{P}[G = 1|X]$. ¿Cómo podemos modelar ésta relación?
- Si usamos un modelo lineal, obtendremos los resultados de la Figura 3, con algunas probabilidades siendo menores a 0.
- Para evitar ésto, usaremos a la *función logística*:

$$p(X) = \frac{\mathrm{e}^{\beta_0+\beta_1 \mathrm{X}}}{1+\mathrm{e}^{\beta_0+\beta_1 \mathrm{X}}} = \frac{1}{1+\mathrm{e}^{-\beta_0-\beta_1 \mathrm{X}}} = \sigma(\beta_0 + \beta_1 X) \tag{4}$$

  - σ es la *función sigmoide*
  - $\lim_{X \to +\infty} p(X) = 1$, $\lim_{X \to -\infty} p(X) = 0$
- A $p(X)/(1 - p(X))$ se les llama las *posibilidades (odds)*.
- El *logit/log-odds* o *posibilidad logarítmica* se define como:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \tag{5}$$

Estimando los coeficientes de regresión

- Podríamos utilizar mínimos cuadrados para encontrar a los parámetros óptimos de 5, pero preferimos utilizar el método de *máxima verosimilitud*.
- Queremos encontrar a $\hat{\beta}_0$ y $\hat{\beta}_1$ tales que $p(X)$ esté lo más cercano posible a 1 para los individuos que incurran en impago y a 0 para aquellos que no.
- Podemos formalizar a ésto mediante la *función de verosimilitud*:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \tag{6}$$

- Por lo tanto:

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0,\beta_1}{\arg\max}\, \ell(\beta_0, \beta_1) \tag{7}$$

Función de máxima verosimilitud

- La *verosimilitud* nos da la probabilidad de los ceros y unos observados en los datos.
- Por ende, queremos maximizar a $\ell(\beta_0, \beta_1)$.
- Si $\hat{\beta}_1 > 0$, entonces un incremento en el `balance` está asociado con un incremento en la probabilidad de incurrir en un `impago`.
- $\hat{\beta}_0$ no nos será de interes en particular; sólo nos servirá a la hora de ajustar las probabilidades a ser las proporciones de unos en los datos.
- La *verosimilitud logarítmica* se define como:

$$\begin{aligned}
\log \ell(\beta_0, \beta_1) &= \log \left(\prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \right) \\
&= \sum_{i:y_i=1} \log p(x_i) + \sum_{i':y_{i'}=0} \log (1 - p(x_{i'})) \\
&= \sum_{i=1}^{n} \{ y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \}
\end{aligned}$$

Código coeficientes de regresión logística (1)

```
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf

df = pd.read_excel("https://goo.gl/qSE6q8")

# .factorize() nos regresa dos objetos: un array con etiquetas
# y uno con valores unicos. Solo nos interesa el primero:
df["default2"] = df.default.factorize()[0]

y = df.default2
X = sm.add_constant(df.balance)

est = smf.Logit(y.ravel(), X).fit()

>>> print(est.summary().tables[1])
```

Código coeficientes de regresión logística (2)

```
import pandas as pd
import sklearn.linear_model as skl_lm

df = pd.read_excel("https://goo.gl/qSE6q8")

df["default2"] = df.default.factorize()[0]

y = df.default2
X = df.balance.values.reshape(-1, 1)

clf = skl_lm.LogisticRegression(solver="newton-cg", C=1e9)
clf.fit(X, y)

>>> print("Clases: ", clf.classes_)
>>> print("Intercepto: ", clf.intercept_)
>>> print("Balance: ", clf.coef_)
```

Haciendo predicciones (1)

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si el `balance` en su tarjeta de crédito es de $\$1000$?
- Ingresamos los coeficientes encontrados:

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | \texttt{balance} = 1000] = \frac{1}{1 + \mathrm{e}^{10.651331 - 0.005499 \times 1000}} = 0.00575 = 0.575\%$$

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si el `balance` en su tarjeta de crédito es de $\$2000$?
- De nuevo:

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | \texttt{balance} = 2000] = \frac{1}{1 + \mathrm{e}^{10.651331 - 0.005499 \times 2000}} = 0.586 = 58.6\%$$

Código coeficientes de regresión logística - predictor cualitativo

```
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf

df = pd.read_excel("https://goo.gl/qSE6q8")

# .factorize() nos regresa dos objetos: un array con etiquetas
# y uno con valores unicos. Solo nos interesa el primero:
df["default2"] = df.default.factorize()[0]
df["student2"] = df.student.factorize()[0]

y = df.default2
X =  sm.add_constant(df.student2)

est = smf.Logit(y.ravel(), X).fit()

>>> print(est.summary().tables[1])
```

Haciendo predicciones (2)

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si es `estudiante`?
- Ingresamos los coeficientes encontrados:

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | \texttt{estudiante} = \texttt{Si}] = \frac{1}{1 + \mathrm{e}^{3.504128 - 0.404887 \times 1}} = 0.0431 = 4.31\%$$

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si no es `estudiante`?
- Ingresamos los coeficientes encontrados:

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | \texttt{estudiante} = \texttt{No}] = \frac{1}{1 + \mathrm{e}^{3.504128 - 0.404887 \times 0}} = 0.0292 = 2.92\%$$

Regresión logística múltiple

- Ahora queremos predecir una respuesta binaria usando múltiples predictores: *regresión logística múltiple*
- Generalizamos a la Ecuación 5 de la siguiente manera:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p \tag{8}$$

donde $X = (X_1, \ldots, X_p)$ son nuestros p predictores.

- Otra forma equivalente sería:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}} = \frac{1}{1+e^{-\beta_0-\beta_1 X_1-\ldots-\beta_p X_p}} \tag{9}$$

- Utilizaremos al estimador de máxima verosimilitud para encontrar a los $\beta_0, \ldots, \beta_p$.

Código coeficientes de regresión logística múltiple

```
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf

df = pd.read_excel("https://goo.gl/qSE6q8")

# .factorize() nos regresa dos objetos: un array con etiquetas
# y uno con valores unicos. Solo nos interesa el primero:
df["default2"] = df.default.factorize()[0]
df["student2"] = df.student.factorize()[0]

y = df.default2
X = sm.add_constant(df[["balance", "income", "student2"]])

est = smf.Logit(y, X).fit()

>>> print(est.summary().tables[1])
```

¿Qué está pasando?

- Ahora el coeficiente de `student` es negativo, indicando que los estudiantes son menos propensos a incurrir en un impago que los no estudiantes.
- Para un `balance` *fijo*, los estudiantes son menos propensos a incurrir en un impago.
- *En general*, los estudiantes tienden a incurrir en un impago más que los no estudiantes.
- ¡Los estudiantes tienden a tener más deuda, lo cual está asociado a una mayor probabilidad de impago!
- Por ende, un estudiante es *más riesgoso* que un no estudiante, si no se tienen sus datos de `balance`.
- Sin embargo, un estudiante es menos riesgoso que un no estudiante *que tenga el mismo* `balance`.
  - A ésto se le conoce como *confounding* (*factor de confusión*)

Confusión en los datos de `Default`

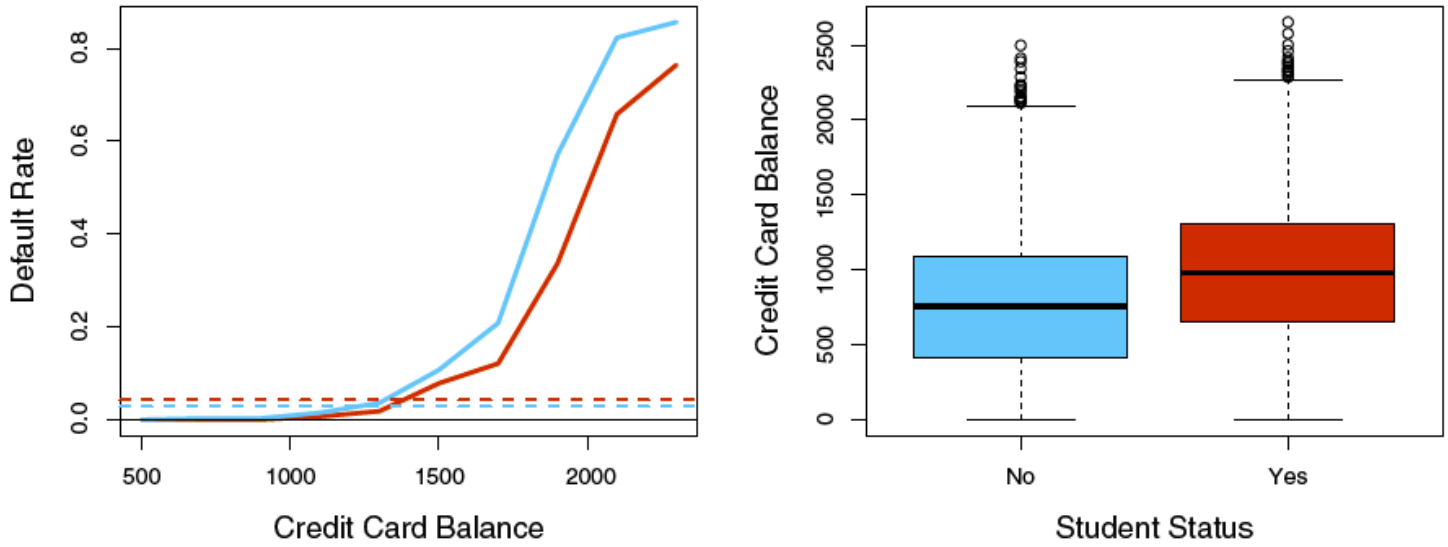

Figura 6: A la izquierda, vemos que las tasas de impago para estudiantes y no estudiantes. La línea sólida representa esta tasa en función del `balance`, mientras que las líneas horizontales punteadas son las tasas de impago generales. A la derecha, tenemos diagramas de caja para estudiantes y no estudiantes. Vemos que los estudiantes tienden a tener un `balance` en la tarjeta de crédito más alto que los no estudiantes, pero para un `balance` fijo, los estudiantes incurren menos en impago.

Haciendo predicciones (3)

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si es `estudiante`, su `balance` sea de $\$1,500$ y su `salario` sea de $\$40,000$?
- Ingresamos los coeficientes encontrados:

$$\begin{aligned}\hat{p}(X) &= \frac{1}{1+\mathrm{e}^{10.869-0.005737\times 1500-0.000003\times 40000+0.6468\times 1}} \\ &= 0.0579 = 5.79\%\end{aligned}$$

- ¿Cuál es la probabilidad de que una persona incurra en un `impago` si no es `estudiante`, su `balance` sea de $\$1,500$ y su `salario` sea de $\$40,000$?
- Ingresamos los coeficientes encontrados:

$$\begin{aligned}\hat{p}(X) &= \frac{1}{1+\mathrm{e}^{10.869-0.005737\times 1500-0.000003\times 40000+0.6468\times 0}} \\ &= 0.1049 = 10.49\%\end{aligned}$$

Regresión Logística Multinomial (1)

- ¿Qué pasa ahora si hay más de dos clases, i.e., si $\mathcal{G} = \{1, \ldots, K\}$?
- Se utiliza la *regresión logística multinomial*.
- Decimos entonces que, para $k = 1, \ldots, K-1$:

$$\mathbb{P}[G = k | X = x] = \frac{\exp\left(\beta_{k0} + \beta_k^\top x\right)}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{l0} + \beta_l^\top x\right)}$$

y para $k = K$:

$$\mathbb{P}[G = K | X = x] = \frac{1}{1 + \sum_{l=1}^{K-1} \exp\left(\beta_{l0} + \beta_l^\top x\right)}$$

Regresión Logística Multinomial (2)

- Nótese que esto induce fronteras de decisión lineales:

$$\{x : \mathbb{P}[G = k|X = x] = \mathbb{P}[G = l|X = x]\}$$

será equivalente a

$$\{x : (\beta_{k0} - \beta_{l0}) + (\beta_k - \beta_l)^\top x = 0\}$$

- Sin embargo, la regresiín logística multinomial no es tan usada en la práctica: es más utilizado el *análisis discriminante*.

Análisis discriminante lineal (LDA)

- En regresión logística, modelamos directamente a $\mathbb{P}[G = k|X = x]$, i.e., modelamos a la distribución condicional de la respuesta G dados los predictores X.
- Veremos otra alternativa: modelar a la distribución de los predictores X para cada una de las clases de respuesta G y luego usar el teorema de Bayes para encontrar a $\mathbb{P}[G = k|X = x]$.
- Si usamos a la distribución normal para cada clase, ésto nos llevará al análisis discriminante lineal o cuadrático.
- Se pueden utilizar otras distribuciones, pero nos enfocaremos en las normales.

¿Por qué necesitamos otro modelo?

- Cuando las clases están claramente separadas, las estimaciones de los parámetros dados por regresión logística son inestables (LDA no sufre de esto).
- Si n es pequeño y la distribución de las clases X es aproximadamente normal, el modelo discriminante lineal es más estable que el de regresión logística.
- LDA es más popular que el model logístico cuando $|\mathcal{G}| > 2$.

Usando el Teorema de Bayes para clasificación (1)

- Suponga que queremos clasificar una observación en una de las K clases ($K \geq 2$).
- Sea π_k la *probabilidad a priori* de que una observación escogida aleatoriamente pertenezca a la clase k.
- Representamos a la *función de densidad de probabilidad* de que la observación X pertenezca a la clase k como

$$f_k(X) \equiv \mathbb{P}[X = x|G = k]$$

  - $f_k(x)$ es relativamente grande si hay una alta probabilidad de que una observación en la clase k tenga $X \approx x$.

Usando el Teorema de Bayes para clasificación (2)

- Por lo tanto, el **teorema de Bayes** dice que:

$$\begin{aligned}\mathbb{P}[G = k|X = x] &= \frac{\mathbb{P}[G = k] \cdot \mathbb{P}[X = x|G = k]}{\mathbb{P}[X = x]} \\ &= \frac{\pi_k f_k(x)}{\sum_{j=1}^{K} \pi_j f_j(x)} = p_k(x)\end{aligned} \tag{10}$$

- Por lo tanto, de envés de calcular directamente a $p_k(x)$, utilizamos a las estimaciones de π_k y $f_k(x)$ en la Ecuación 10.
- Nos referimos a $p_k(x)$ como la *probabilidad a posteriori*.

Usando el Teorema de Bayes para clasificación (3)

- Podemos estimar a π_k fácilmente: calculamos la fracción de los datos de entrenamiento que pertenezcan a la clase k.
  - Se debe de cumplir que $\sum_{k=1}^{K} \pi_k = 1$.
- Estimar a $f_k(x)$ es más complicado.
- Si logramos encontrar una manera de estimar a $f_k(x)$, podemos desarrollar a un clasificador que aproxime al clasificador de Bayes.

Análisis Discriminante Lineal para $p = 1$ (1)

- Asumimos la forma de $f_k(x)$ como una función de densidad de una *distribución normal* o *Gaussiana*:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right) \tag{11}$$

  donde μ_k y σ_k^2 son los parámetros de media y varianza para la clase k.
- Asumamos, por ahora, que $\sigma_1^2 = \cdots = \sigma_K^2 = \sigma^2$.
- Reemplazando a la Ecuación 11 en la Ecuación 10, llegamos a:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{j=1}^{K} \pi_j \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_j)^2\right)} \tag{12}$$

Análisis Discriminante Lineal para $p = 1$ (2)

- Vemos lo siguiente:

$$\begin{aligned}
\log\left(\frac{p_k(x)}{p_l(x)}\right) &= \log\left(\frac{\pi_k \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\pi_l \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) - \frac{1}{2\sigma^2}(x-\mu_k)^2 + \frac{1}{2\sigma^2}(x-\mu_l)^2 \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) - \frac{x^2}{2\sigma^2} + \frac{x\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \frac{x^2}{2\sigma^2} - \frac{x\mu_l}{\sigma^2} + \frac{\mu_l^2}{2\sigma^2} \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + x \cdot \frac{\mu_k - \mu_l}{\sigma^2} - \frac{\mu_k^2 - \mu_l^2}{2\sigma^2}
\end{aligned}$$

  - ¡Una función lineal!

Análisis Discriminante Lineal para $p = 1$ (3)

- Definimos a nuestra *función discriminante lineal* como:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k \tag{13}$$

- Vemos que nuestra regla de decisión se puede escribir de manera equivalente como:

$$G(x) = \arg\max_k \delta_k(x)$$

- Nótese, entonces, que $\log\left(\frac{p_k(x)}{p_l(x)}\right) = \delta_k(x) - \delta_l(x)$
- Si $K = 2$ y $\pi_1 = \pi_2$, entonces la frontera de decisión de Bayes corresponde al punto donde

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

Clasificamos a la densidad más alta

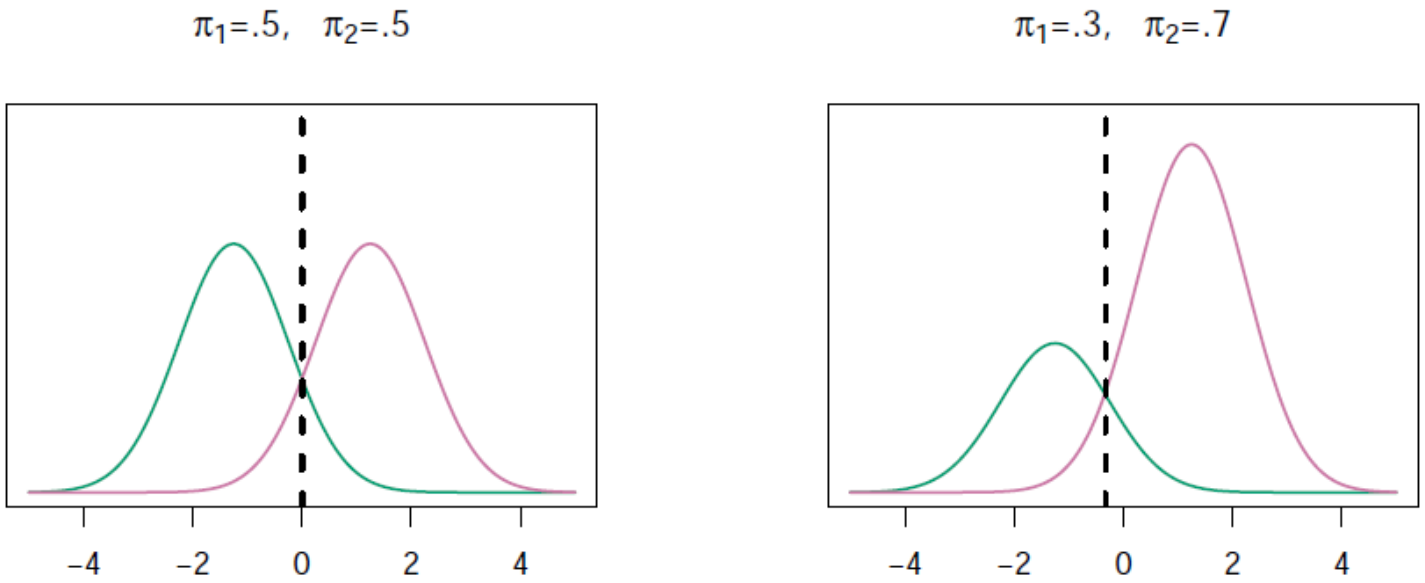

Figura 7: Clasificamos a un dato de acorde con cuál $\pi_k f_k(x)$ es más alto o, de manera equivalente, cuál discriminante lineal $\delta_k(x)$ es más alto. A la derecha, favorecemos a la clase morada, por lo que la frontera de decisión se ha corrido a la izquierda. En ambos casos, tendremos que $\mu_1 = -1.25$, $\mu_2 = 1.25$ y $\sigma_1 = \sigma_2 = 1$.

Análisis Discriminante Lineal para $p = 1$ (4)

- En la vida real, aunque estemos seguros de nuestra suposición de que X tiene una distribución normal ($X \sim \mathcal{N}(\mu, \sigma^2)$), aún tendremos que estimar a los parámetros $\mu_1, \ldots, \mu_K$, $\pi_1, \ldots, \pi_K$ y a σ^2.
- El método de *Análisis Discriminante Lineal (LDA)* aproxima al clasificador de Bayes al ingresar, en la Ecuación 13, estimaciones de dichos parámetros.
- Utilizaremos las siguientes:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:g_i=k} x_i \tag{14}$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:g_i=k} (x_i - \hat{\mu}_k)^2 \tag{15}$$

donde $n = |\mathcal{T}_{\mathsf{Tr}}|$ (el número total de observaciones de entrenamiento) y n_k es el número de observaciones de entrenamiento en la clase k.

Análisis Discriminante Lineal para $p = 1$ (5)

- A veces se tiene conocimiento de las probabilidades a priori $\pi_1, \ldots, \pi_K$, por lo que podemos utilizarlas directamente.
- De lo contrario, usamos a las proporciones en los datos de entrenamiento por clase:

$$\hat{\pi}_k = n_k/n \tag{16}$$

- Por lo tanto, el *clasificador LDA* asigna a una observación $X = x$ la clase para la cual la función discriminante

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log \hat{\pi}_k \tag{17}$$

es mayor, i.e., $G = \arg\max_k \hat{\delta}_k(x)$.

Frontera de decisión del LDA

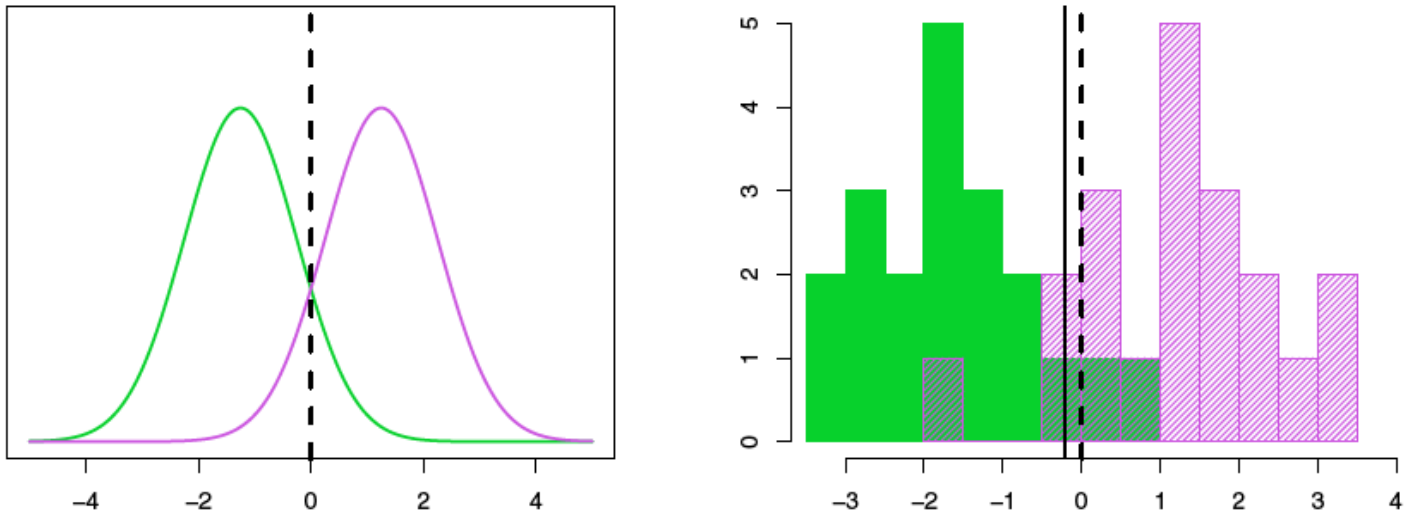

Figura 8: En ambas gráficas, la línea vertical punteada nos muestra la frontera de decisión de Bayes, $x = (\mu_1 + \mu_2)/2$, para dos funciones de densidad normales, con $\mu_1 = -1.25$, $\mu_2 = 1.25$, $\sigma_1^2 = \sigma_2^2 = 1$ y $\pi_1 = \pi_2 = 0.5$. A la derecha, hemos tomado 20 observaciones de cada clase ($n_1 = n_2 = 20$) y mostramos el histograma. La línea negra sólida representa la frontera de decisión del LDA que ha sido estimado de los datos, i.e., $x = (\hat{\mu}_1 + \hat{\mu}_2)/2$. El error de Bayes es de 10.6% y el error del LDA es de 11.1%.

Análisis Discriminante Lineal para $p > 1$ (1)

- Tendremos ahora que $X = (X_1, \ldots, X_p)$ tiene una *distribución normal (Gausiana) multivariante*: $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, donde $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)$ y $\boldsymbol{\Sigma}$, la matriz $p \times p$ de covarianza de X:

$$\boldsymbol{\Sigma} = \text{Cov}(X) = \mathbb{E}\left[XX^\top\right] - \boldsymbol{\mu}\boldsymbol{\mu}^\top$$

- En otras palabras, asumimos que cada predictor sigue una distribución normal unidimensional, con alguna correlación entre cada par de predictores X_i y X_j dada por $\boldsymbol{\Sigma}_{ij}$.
- La función de densidad está dada por:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(x - \boldsymbol{\mu})\right) \tag{18}$$

Distribución normal multivariante con $p = 2$

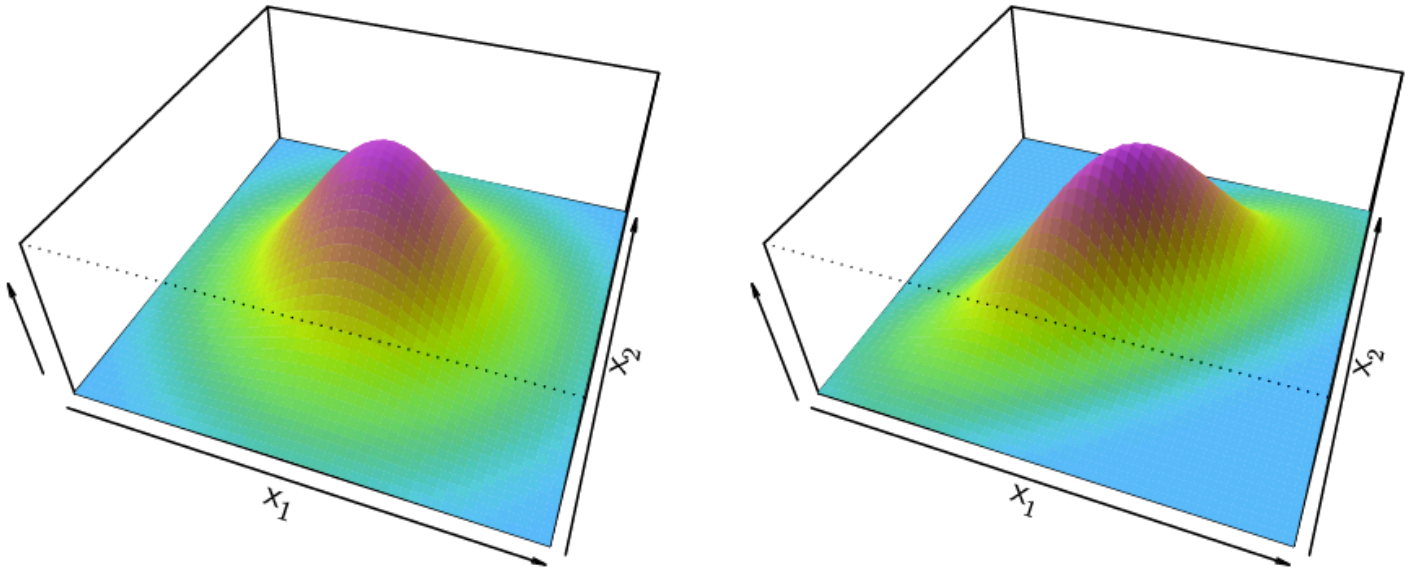

Figura 9: Dos funciones de densidad normal multivariantes, con $p = 2$. A la izquierda, los predictores X_1 y X_2 tienen misma varianza, $\mathbb{V}[X_1] = \mathbb{V}[X_2]$ y no están correlacionados, por lo que $\text{Cor}(X_1, X_2) = 0$. A la derecha, los predictores tienen una correlación de 0.7. La base de la campana ya no es circular, sino elíptica.

Análisis Discriminante Lineal para $p > 1$ (2)

- En otras palabras, las observaciones en la clase k se extraen de una distribución normal multivariante $\mathcal{N}(\mu_k, \mathbf{\Sigma})$.
- El clasificador de Bayes asigna una observación $X = x$ a la clase para la cual la función discriminante

$$\delta_k(x) = x^\top \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^\top \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k \tag{19}$$

  es mayor, i.e., $G = \arg\max_k \delta_k(x)$.
- La versión matricial de la Ecuación 13.
- Sigue siendo una función lineal.
- Necesitaremos aproximaciones a los parámetros, por lo que utilizamos las versiones matriciales de las Ecuaciones 14, 15 y 16.

Dos predictores $p = 2$ y $K = 3$ clases

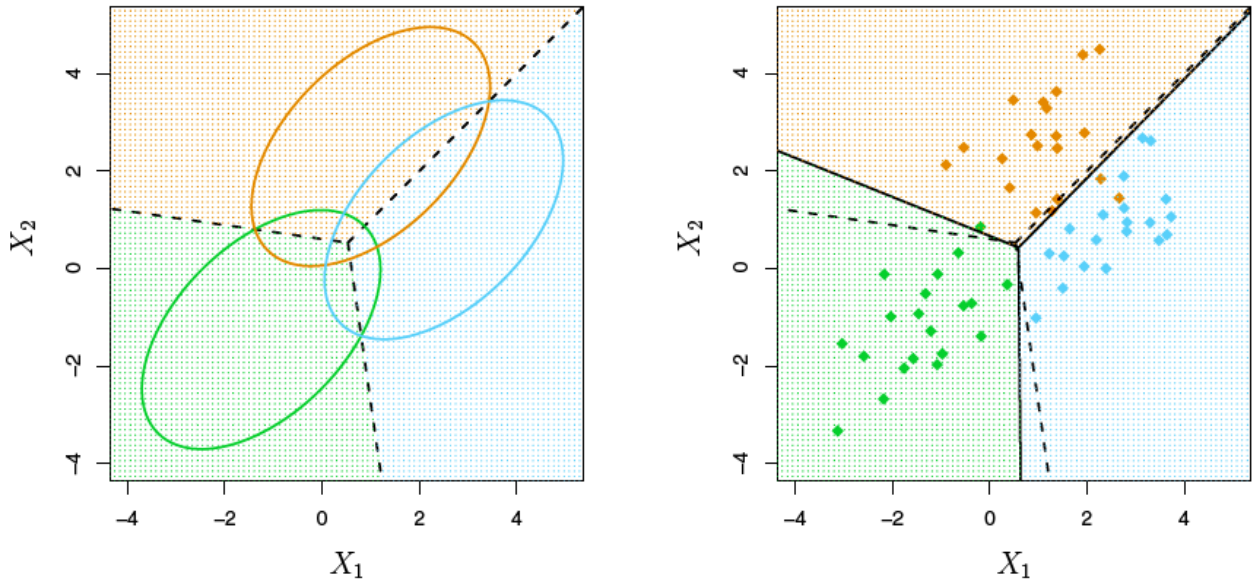

Figura 10: Se muestran tres clases Gausianas de igual tamaño, con un vector de la media específico por clase, así como una matriz de covarianza común para las tres clases. Las elipses representan las regiones que contienen el 95% de los datos de cada una de las clases. Las líneas punteadas son las fronteras de decisión de Bayes, i.e., donde $\delta_k(x) = \delta_l(x)$. Tendremos que $\pi_1 = \pi_2 = \pi_3 = 1/3$. A la derecha se han generado 20 observaciones por clase y se muestran las fronteras de decisión del LDA con una línea negra sólida. El error de prueba de Bayes es de 0.0746 y el del LDA es de 0.0770.

Llevándolo a probabilidades

- Otra manera de facilitar la interpretación de las funciones discriminantes $\hat{\delta}_k(x)$ es transformando los resultados a probabilidades.
- Lo logramos mediante la *función Softmax*:

$$\hat{\mathbb{P}}[G = k|X = x] = \frac{\exp \hat{\delta}_k(x)}{\sum_{l=1}^{K} \exp \hat{\delta}_l(x)} \tag{20}$$

- Así, normalizamos a los valores obtenidos para que estén en el rango $[0, 1]$ y que su suma sea igual a 1.
- Cuando $p = 2$, lo usual es asignar a una clase si la probabilidad es mayor a 0.5, i.e., si $\hat{\mathbb{P}}[G = k|X = x] \geq 0.5$.
- Se pueden usar otros límites, dependiendo de la aplicación y qué tan conservador es uno.

Código LDA, $p = 2$ y $K = 2$

```
import pandas as pd
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

df = pd.read_excel("https://goo.gl/qSE6q8")

# .factorize() nos regresa dos objetos: un array con etiquetas
# y uno con valores unicos. Solo nos interesa el primero:
df["default2"] = df.default.factorize()[0]
df["student2"] = df.student.factorize()[0]

y = df.default2.values
X = df[["balance", "student2"]].values

lda = LinearDiscriminantAnalysis().fit(X, y)
y_pred = lda.predict(X)

>>> print("Error: {:.2f}".format(100*(1 - sum(y == y_pred)/len(y))), "%")
>>> print("Error: {:.2f}".format(100*(1 - lda.score(X, y))), "%")[1]
```

[1] ¿Es un buen modelo?

Análisis del LDA

- El error del LDA es de 2.75%, pero debemos de tener en mente dos cosas:
  - Estamos hablando de error de entrenamiento, por lo que mientras más alta es la razón p/n, es más probable que estemos *sobreajustando*. No es el caso aquí, ya que $p = 2$ y $n = 10,000$.
  - Ya que sólo el 3.33% de los individuos en los datos incurrieron en un impago un *clasificador nulo* que clasifica a todos como `No` haría un trabajo comparable con nuestro clasificador LDA.
  - Podemos conseguir lo último mediante `100*sum(y)/len(y)`.
- Podemos realizar dos tipos de error: asignarle incorrectamente a un individuo que incurre en un impago la categoría `No` (*Error tipo I*) o asignarle incorrectamente a un individuo que no incurre en un impago la categoría `Si` (*Error tipo II*).

Métricas a seguir (1)

- Generalmente deseamos saber qué tipo de error estamos cometiendo.
- Utilizamos una *matriz de confusión* para desplegar esta información.
- Para una respuesta binaria, tendremos cuatro elementos en nuestra matriz de confusión:
  - *Verdadero negativo (TN)* cuando $\hat{G} = \texttt{No}$ y $G = \texttt{No}$.
  - *Verdadero positivo (TP)* cuando $\hat{G} = \texttt{Si}$ y $G = \texttt{Si}$ (*potencia*).
  - *Falso negativo (FN)* cuando $\hat{G} = \texttt{No}$ y $G = \texttt{Si}$ (error tipo II).
  - *Falso positivo (FP)* cuando $\hat{G} = \texttt{Si}$ y $G = \texttt{No}$ (error tipo I).

Métricas a seguir (2)

- Con éstos, podemos definir a otras tres métricas:
  - *Sensibilidad o exhaustividad (True Positive Rate, TPR)*: el porcentaje de individuos que realmente incurren en un impago que son correctamente identificados:

$$\text{Sensibilidad} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \text{TPR} = \text{recall}$$

  - *Especificidad (True Negative Rate, TNR)*: el porcentaje de individuos que no incurren en un impago que son correctamente identificados:

$$\text{Especificidad} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \text{TNR}$$

  - *Precisión*: el porcentaje de individuos que incurren en un impago que son identificados:

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Matriz de confusión

```
from sklearn.metrics import confusion_matrix
y_actual = pd.Series(y, name="Actual")
y_predicted = pd.Series(y_pred, name="Predicho")

y_actual = y_actual.replace([0, 1], ["No", "Si"])
y_predicho = y_predicho.replace([0, 1], ["No", "Si"])

>>> pd.crosstab(y_actual, y_predicho, margins=True)

>>> print(confusion_matrix(y_actual, y_pred))

tn, fp, fn, tp = confusion_matrix(y, y_pred).ravel()
```

Métricas a seguir (3)

- Definimos al *error total cometido* como $(\mathsf{FP} + \mathsf{FN})/n$
- La *tasa de falsos negativos (FNR)* es la proporción de individuos que incurrieron en un impago que fueron clasificados como `No`:

$$\mathsf{FNR} = \frac{\mathsf{FN}}{\mathsf{FN} + \mathsf{TP}}$$

- La *tasa de falsos positivos (FPR)* es la proporción de individuos que no incurrieron en un impago que fueron clasificados como `Si`:

$$\mathsf{FPR} = \frac{\mathsf{FP}}{\mathsf{FP} + \mathsf{TN}}$$

- El *valor F balanceado (F1)* es la media harmónica entre precisión y exhaustividad:

$$F_1 = \frac{2}{\text{precisión}^{-1} + \text{exhaustividad}^{-1}} = \frac{2\mathsf{TP}}{2\mathsf{TP} + \mathsf{FP} + \mathsf{FN}}$$

Modificando al umbral

- Una compañía de tarjetas de crédito puede estar más interesada en evitar el Error tipo II que el Error tipo I.
- En otras palabras, le es preferible no dar una tarjeta de crédito a un individuo que no va a incurrir en un impago que darsela a uno que sí va a incurrir en un impago.
- Podemos, entonces modificar al umbral de decisión para clasificar a los individuos/clientes como aquellos que `Si` incurrirán en un impago a uno más bajo, i.e., pasar de

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | X = x] > 0.5 \tag{21}$$

a

$$\hat{\mathbb{P}}[\texttt{impago} = \texttt{Si} | X = x] > 0.2 \tag{22}$$

- El Error tipo II baja de $252/333 = 75.7\%$ a $138/333 = 41.4\%$, mientras que el Error tipo I sube de $23/9667 = 0.238\%$ a $235/9667 = 2.43\%$. A la compañía le puede parecer aceptable.

Código para Métricas y Umbral

Link al código: https://goo.gl/vPNmpB

- Vemos que podemos seguir modificando el umbral y ver el intercambio que sucede entre las métricas.
- Seleccionar el umbral correcto depende del *conocimiento del campo* (e.g., los costos incurridos al cometer los tipos de errores).
- El error total baja pero la FNR sube.

Curva ROC

- Una *curva ROC* muestra el desempeño de un clasificador con todos los umbrales posibles al comparar la tasa de falsos positivos versus la tasa de verdaderos positivos.
- Podemos resumir al clasificador por el *área bajo la curva (AUC)*.
- Una curva ROC ideal se aferrará a la esquina superior izquierda, por lo que AUC $= 1$.
- Un clasificador que no es mejor que una decisión aleatoria tendrá AUC $= 0.5$, o un ROC que es una línea a $45°$.

Curva ROC para los datos de Default

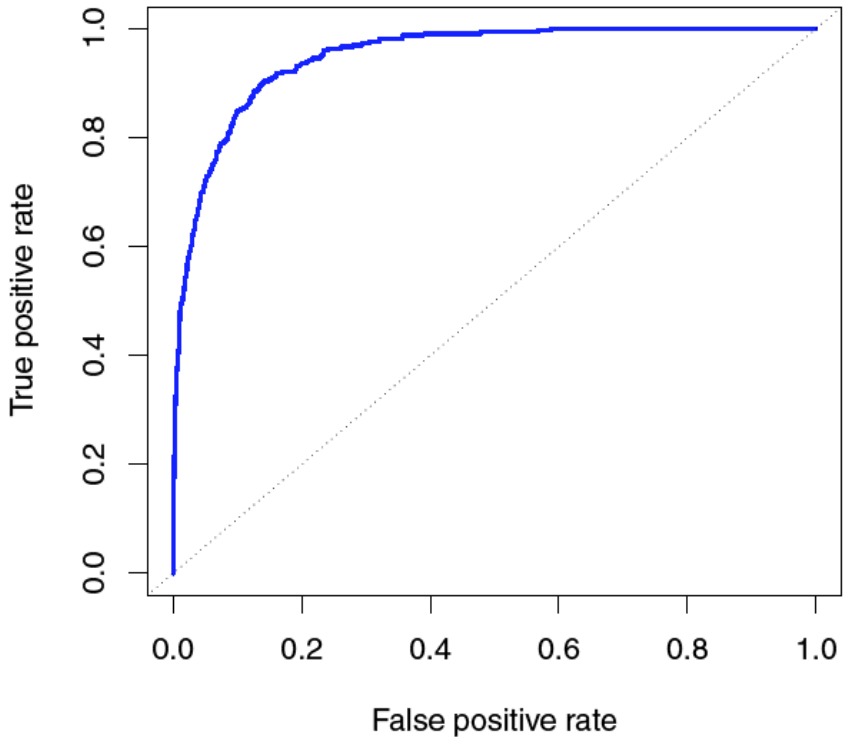

Figura 11: La curva ROC para el clasificador LDA en los datos de Default. Conforme modifiquemos el umbral, cambiarán tanto el FPR como el TPR. La línea punteada es el clasificador que no es mejor que una decisión aleatoria. Por lo tanto, deseamos que nuestro clasificador esté siempre por arriba de dicha línea.

Análisis Discriminante Cuadrático (QDA)

- El *análisis discriminante cuadrático (QDA)* tiene las mismas suposiciones que el LDA, excepto que ahora asumimos que cada clase tiene su propia matriz de covarianza $\boldsymbol{\Sigma}_k$.
- Por lo tanto, dichos términos no van a desaparecer y definimos a la *función discriminante cuadrática*:

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(x-\boldsymbol{\mu}_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^\top \boldsymbol{\Sigma}_k^{-1}x + x^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log\pi_k\end{aligned} \tag{23}$$

- La frontera de decisión $\{x : \delta_k(x) = \delta_l(x)\}$ serán ecuaciones cuadráticas.
- Nuestra decisión será nuevamente: $G = \arg\max_k \delta_k(x)$. De manera equivalente, podemos calcular las probabilidades utilizando a la Ecuación 20 (función Softmax).

LDA vs. QDA para dos casos

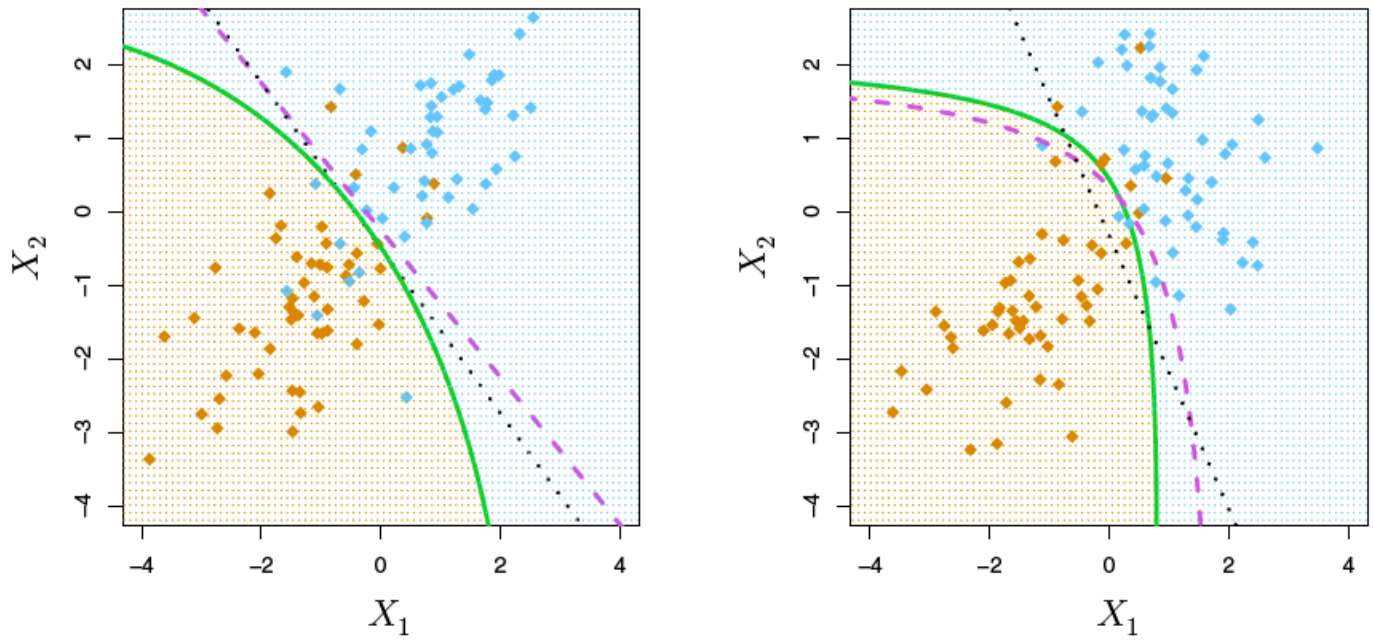

Figura 12: Presentamos a la frontera de decisión del LDA es la línea negra punteada; la del QDA es la línea verde y la de Bayes es la línea punteada morada. A la derecha, tendremos datos generados con $\Sigma_1 = \Sigma_2$ y podemos apreciar que la frontera de decisión de Bayes es lineal, por lo que LDA la aproxima mejor. A la izquerda, tendremos que $\Sigma_1 \neq \Sigma_2$ y la frontera de decisión de Bayes es no lineal, por lo que QDA la aproxima mejor.

LDA vs. QDA

- Cuando tenemos p predictores, estimar la matriz de covarianza requiere de estimar a $p(p+1)/2$ parámetros.
- Por lo tanto, ya que QDA requiere de K distintas covarianzas, se tendrán que estimar $Kp(p+1)/2$ parámetros.
- En cambio, LDA es un modelo lineal, por lo que se requieren estimar a Kp parámetros.
- Por lo tanto, LDA es mucho menos flexible que QDA y tiene menor varianza, pero mayor sesgo.
- LDA realiza un mejor trabajo que QDA si hay pocos datos de entrenamiento; se recomienda usar a QDA si el conjunto de entrenamiento es muy grande y la varianza del clasificador no es una preocupación.

KNN

- El *clasificador de k vecinos cercanos (KNN)* no asume la distribución de los datos.
- Primero identifica a los K puntos más cercanos a una observación de prueba x_0, los cuales representamos por $\mathcal{N}_0$.
- Estimamos a la probabilidad condicional de que dicho punto pertenezca a la clase j como:

$$\mathbb{P}[G = j | X = x_0] = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{1}_{g_i = j}(x_0) \tag{24}$$

- La elección de K tiene efectos dramáticos en el clasificador obtenido.

Ilustración de KNN

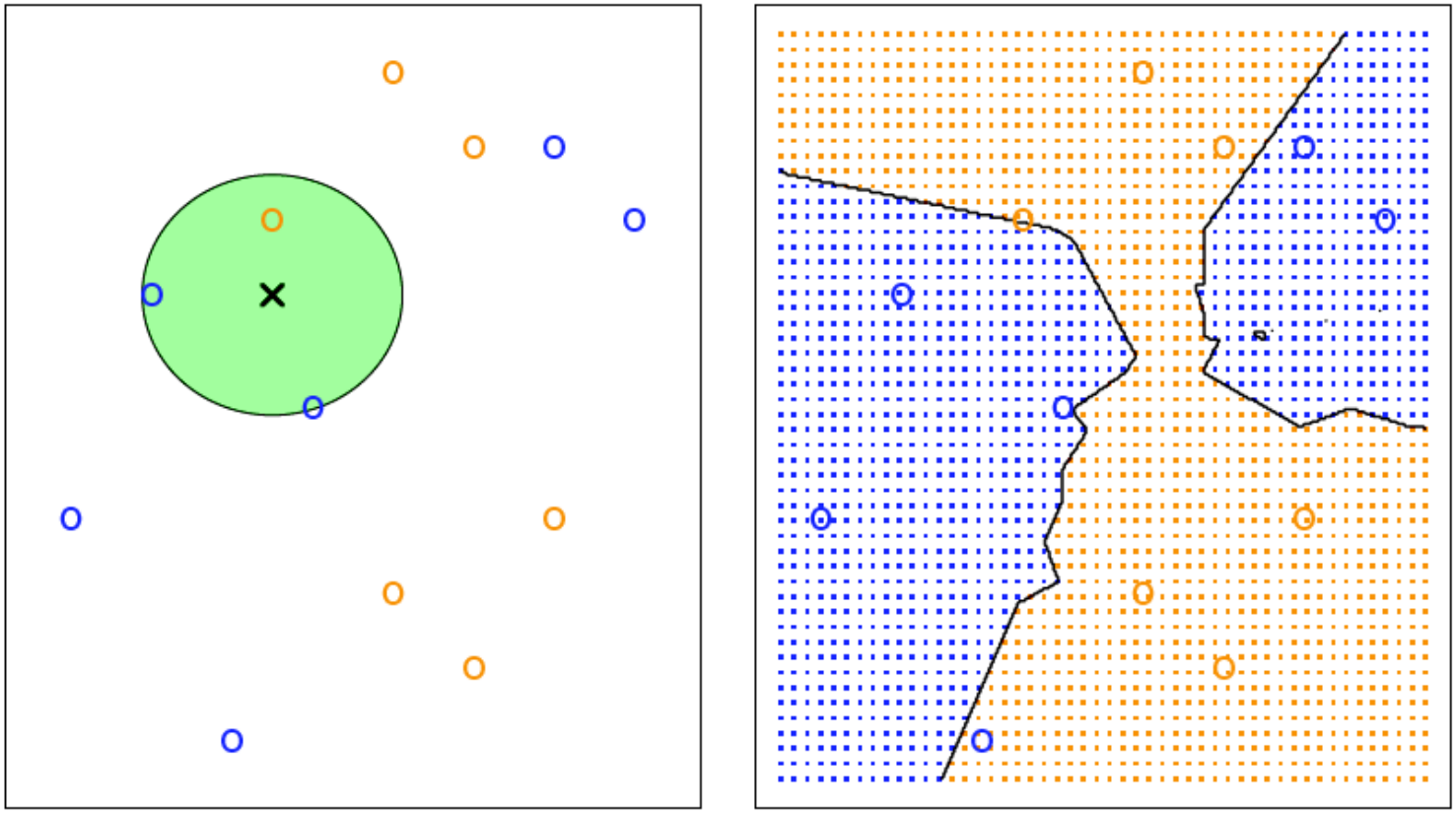

Figura 13: Usando a $K = 3$, el círculo verde nos indica que, para clasificar a un punto que se localiza en la cruz, debemos de tomar en consideración a los dos puntos azules y al punto anaranjado. A la derecha mostramos las fronteras de decisión generadas con los datos y número de vecinos cercanos.

Código KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)

knn.fit(X, y)

y_predicho = knn.predict(X)

>>> print(confusion_matrix(y, y_pred))

tn, fp, fn, tp = confusion_matrix(y, y_pred).ravel()
```

Variando a K en KNN

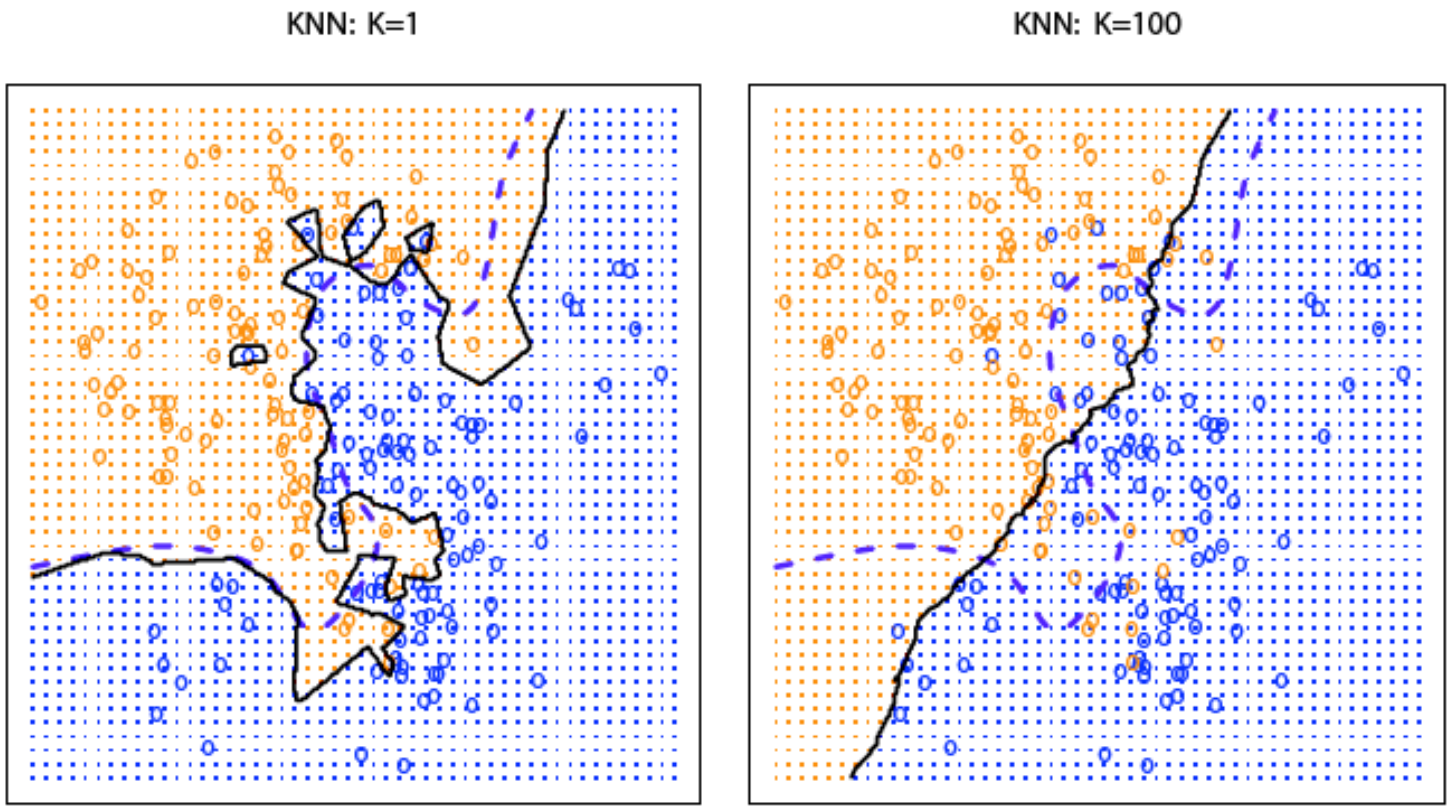

Figura 14: A la izquierda, usamos $K = 1$ vecinos cercanos, mientras que a la derecha usamos $K = 100$ vecinos cercanos para datos generados. La frontera de decisión de Bayes es la línea punteada. Nótese que, conforme aumentemos a K, la frontera de decisión de KNN se vovlerá cada vez más lineal (menos flexible).

Últimas notas

- Existe otro tipo de función discriminante llamado *Naive Bayes*.
- Es útil cuando tenemos predictores cualitativos y cuantitativos, así como cuando p es muy grande.
- Tendremos que:

$$\delta_k(x) \propto \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right]$$
$$= -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log\sigma_{kj}^2\right] + \log\pi_k$$

- A pesar de sus suposiciones fuertes, usualmente produce buenos resultados.