

Ejercicio 1

La meta de esta tarea es explorar un otro clasificador : El clasificador por regresión logística. La regresión logística se trata de regresar una función de probabilidad de pertenencia condicional $p_i(x) = \mathbb{P}(Y = i|X = x)$ de la forma

$$p_i(x) \approx \sigma(a_i^T X + a_{i,0}),$$

donde σ es un mapeo continuo y creciente de \mathbb{R} a $[0, 1]$. En general se toma $\sigma(t) = 1/(1 + e^{-t})$.

1. Suponiendo que tenemos acceso a las regresiones logísticas de los p_i , describir el clasificador óptimo g basado en los \hat{p}_i .
2. Mostrar que las fronteras entre las diferentes clases son sub-conjuntos de hiperplanos (Es un clasificador lineal).
3. Usar el script siguiente para cargar los datos de iris.

```
from sklearn import datasets
iris = datasets.load_iris()
X = iris.data[:, :2]
Y = iris.target
```

4. De la librería sklearn, usar las funciones DecisionBoundaryDisplay para visualizar las R_i de la clasificación y LogisticRegression para definir el modelo de clasificador por regresión logística.

```
from sklearn.inspection import DecisionBoundaryDisplay
from sklearn.linear_model import LogisticRegression
```

Producir un plot que permite visualizar las regiones R_i del clasificador. Superponer los datos de entrenamiento.

5. Cuantos datos están mal clasificados?

Desarrollemos la expresión de la probabilidad condicional, en esta caso tenemos para dos clases $\mathcal{Y} = \{0, 1\}$. Tenemos

$$\begin{aligned}\mathbb{P}(Y = 1|X = x) &= \frac{1}{1 + e^{-(a_{1,0} + a_1^T x)}}, \\ &= \frac{e^{a_{1,0} + a_1^T x}}{1 + e^{a_{1,0} + a_1^T x}}\end{aligned}$$

y

$$\begin{aligned}\mathbb{P}(Y = 0|X = x) &= \frac{1}{1 + e^{-(a_{0,0} + a_0^T x)}}, \\ &= \frac{e^{a_{0,0} + a_0^T x}}{1 + e^{a_{0,0} + a_0^T x}}\end{aligned}$$

donde $\mathbb{P}(Y = 1|X = x) + \mathbb{P}(Y = 0|X = x) = 1$.

Dado que la función inversa de $\rho(t)$ obtenida por

$$\begin{aligned}\rho(t) &= \frac{e^t}{1 + e^t}, \\ 1 + e^{-t} &= \frac{1}{\rho(t)}, \\ e^{-t} &= \frac{1 - \rho(t)}{\rho(t)}, \\ t &= \log \left(\frac{\rho(t)}{1 - \rho(t)} \right)\end{aligned}$$

entonces podemos escribir

$$\log \left(\frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = 0|X = x)} \right) = a_{1,0} + a_1^T x$$

En el caso genereal, a más de dos clases $\mathcal{Y} = \{1, 2, \dots, K\}$ la clasificación por regresión logística se generaliza a considerar las probabilidades condicional como

$$\begin{aligned}\log \left(\frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = K|X = x)} \right) &= a_{1,0} + a_1^T x, \\ \log \left(\frac{\mathbb{P}(Y = 2|X = x)}{\mathbb{P}(Y = K|X = x)} \right) &= a_{2,0} + a_2^T x, \\ &\vdots \\ \log \left(\frac{\mathbb{P}(Y = K - 1|X = x)}{\mathbb{P}(Y = K|X = x)} \right) &= a_{K-1,0} + a_{K-1}^T x,\end{aligned} \tag{1}$$

donde se impone la condición

$$1 = \mathbb{P}(Y = 1|X = x) + \dots + \mathbb{P}(Y = K - 1|X = x) + \mathbb{P}(Y = K|X = x)$$

y manipulando algebraicamente

$$\frac{1}{\mathbb{P}(Y = K|X = x)} = \frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = K|X = x)} + \frac{\mathbb{P}(Y = K - 1|X = x)}{\mathbb{P}(Y = K|X = x)} + 1$$

y del modelo logístico

$$\frac{1}{\mathbb{P}(Y = K|X = x)} = e^{a_{1,0} + a_1^T x} + e^{a_{2,0} + a_2^T x} + \dots + e^{a_{K-1,0} + a_{K-1}^T x} + 1$$

por tanto

$$p_K(x) = \mathbb{P}(Y = K|X = x) = \frac{1}{1 + \sum_{i=1}^{K-1} e^{a_{i,0} + a_i^T x}}$$

y de (1) se obtiene

$$p_j(x) = \mathbb{P}(Y = j|X = x) = \frac{e^{a_{j,0} + a_j^T x}}{1 + \sum_{i=1}^{K-1} e^{a_{i,0} + a_i^T x}}$$

para $j = 1, \dots, K - 1$.

Así, recordando que la regla óptima de clasificación es

$$g(x) = \operatorname{argmax}_j \mathbb{P}(Y = j|X = x)$$

al tener acceso a la regresión logística \hat{p}_j , entonces el estimador de clasificador óptimo es

$$\hat{g}(x) = \operatorname{argmax}_j \hat{p}_j(x) \quad (2)$$

donde usualmente los $\hat{p}_j(x)$ se obtiene numericamente.

Para probar que la frontera es un subconjunto de un hiperplano, consideremos dos clases $i, j \in \mathcal{Y}$. Sin pérdida de generalidad $i \neq K$ y $j \neq K$. Entonces, la regla de clasificación nos dice que clasificaremos $x \in \mathbb{R}^n$ en i si $p_i(x) > p_j(x)$ para $j \neq i$

$$p_i(x) > p_j(x)$$

usando la estimación por regresión logística

$$\begin{aligned} \hat{p}_i(x) &> \hat{p}_j(x), \\ \frac{1}{1 + e^{-(\hat{a}_{i,0} + \hat{a}_i^T x)}} &> \frac{1}{1 + e^{-(\hat{a}_{j,0} + \hat{a}_j^T x)}}, \\ 1 + e^{-(\hat{a}_{j,0} + \hat{a}_j^T x)} &> 1 + e^{-(\hat{a}_{i,0} + \hat{a}_i^T x)}, \\ -(\hat{a}_{j,0} + \hat{a}_j^T x) &> \hat{a}_{i,0} + \hat{a}_i^T x, \\ \hat{a}_{i,0} - \hat{a}_{j,0} + (\hat{a}_i - \hat{a}_j)^T x &> 0 \end{aligned}$$

que forma un subconjunto de un hiperplano, como deseaba mostrarse.

Ejercicio práctico

Tenemos la base de datos que se conforma de una serie de 50 datos por tres clases diferentes de plantas para el largo y ancho de su sépalo y largo y ancho de su pétalo. En este ejercicio nos restringiremos a estudiar solamente el largo y ancho del sépalo ya que es más ilustrativo.

Ahora, después de cargar los datos, tenemos que importar las paqueterías mencionadas en el enunciado. Con *LogisticRegression* nos permite ajustar un modelo logístico de tres clases a los datos. Con la paquetería *DecisionBoundaryDisplay* nos permite graficar la regla de clasificación en sus diferentes regiones y fronteras.

Como es buena práctica para probar la eficiencia del modelo ajustado, tomamos una partición de la base de datos para entrenamiento y prebas. Con el comando

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.20, random_state=0)
```

Luego, el ajuste con los datos de entrenamiento se construye con

```
logisticRegr = LogisticRegression()
clasificador = logisticRegr.fit(x_train, y_train)
```

Posteriormente, para usar la paquetería es necesario predecir el valor en cada punto de la región de interés. Por tanto, se construye una malla que va de los extremos de los regresores, X , y se pide al modelo que prediga que valor toma en cada punto, es decir se aplica la regla de clasificación.

La gráfica de las regiones queda vista a continuación

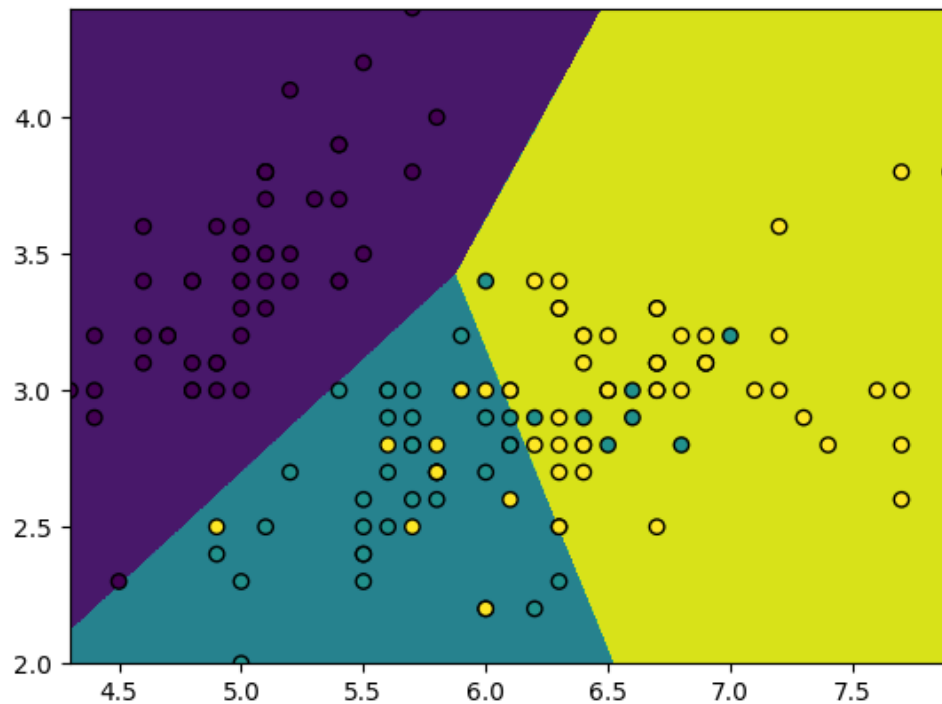


Figura 1: Regiones de clasificación para regresión logística de la base de datos Iris para los sépalos de tres clases diferentes de plantas.

Notemos que las fronteras son efectivamente lineales, como se provó en un ejercicio previo. Además vemos que el ajuste con las clases en morado son excelentes, sin embargo ahí donde las clases se junta (amarillo y azul) tenemos clasificadores erróneos.

Finalmente para probar la precisión del modelo, consideremos el comando

```
score = logisticRegr.score(x_test, y_test)
print("Proporción de veces que acierta el modelo: ", score)
```

que nos dice la proporción de datos bien clasificados contra el número total de datos. Vemos que al ejecutar el script para la partición de prueba del 20 % tenemos que hay 73.333 % de datos bien clasificados, es decir 110 datos de los 150 clasificó el modelo de manera asertiva. Luego 40 de los 150 fueron erróneos.

El **enlace al código** con los detalles del ejercicio práctico se encuentra [aquí](#).