

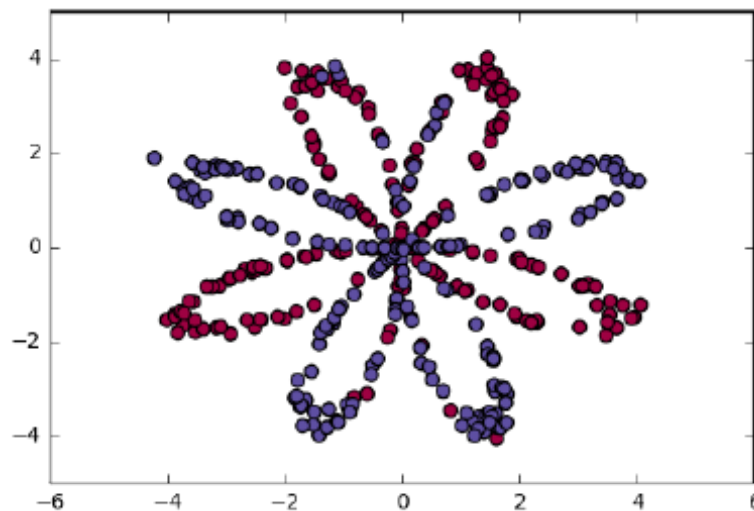


Inteligencia Artificial  
Grado en Ingeniería Informática en Sistemas de Información - Curso 2019/20  
**HOMEWORK #2: Regresión Logística**

Bienvenidos a la segunda tarea de la asignatura Inteligencia Artificial. Los modelos de aprendizaje de regresión logística presentan una gran flexibilidad para separar conjuntos de datos mediante fronteras no lineales. En esta tarea aprenderás a usar la regresión logística para separar dos clases que no se pueden separar bien a través de una línea recta.

### Problema 1

En este problema implementaremos una regresión logística para el conjunto de datos que se muestra en la figura.



1. Cargue y visualice el conjunto de datos usando la función `plotData`.
2. Implementar una regresión logística. En este caso no se hará una evaluación del modelo mediante un conjunto de test sino que simplemente queremos analizar como la frontera de decisión aprendida a partir de todo el conjunto de datos separa las dos clases representadas por color azul y rojo. Para ello, obtenga un modelo usando todo el conjunto de datos y haga una predicción de todo el conjunto de datos, imprimiendo por pantalla la tasa de acierto y la gráfica con la frontera de decisión usando para ello la función `plotDecisionBoundary`.

**Nota:** No se debe usar ninguna función de optimización avanzada sino que debe implementar el método de descenso del gradiente para regresión logística.

Con el objeto de verificar el código tenga en cuenta las siguientes soluciones parciales:

Función `coste`:

```
Cost at initial theta (zeros): 0.693147
Gradient at initial theta (zeros):
-0.000000
-0.022562
0.196977
```

Función `descenso del gradiente` para `num_iter=10` y `alfa=1`

```
Theta at iteration 1: 0.000000 0.022562 -0.196977
Theta at iteration 2: 0.003571 0.027415 -0.202265
Theta at iteration 3: 0.006426 0.027885 -0.202883
Theta at iteration 4: 0.008611 0.027964 -0.202996
Theta at iteration 5: 0.010274 0.027997 -0.203045
Theta at iteration 6: 0.011538 0.028020 -0.203078
Theta at iteration 7: 0.012500 0.028037 -0.203104
Theta at iteration 8: 0.013232 0.028050 -0.203123
```



```
Theta at iteration 9: 0.013788 0.028060 -0.203138  
Theta at iteration 10: 0.014211 0.028067 -0.203149
```

3. Como se puede comprobar en la figura del apartado 2, los datos no son separables linealmente y se necesita una frontera de decisión mucho más compleja que una simple recta. Por tanto, el modelo tiene que aprender los patrones de las hojas de la flor. Una solución es crear más atributos mediante la función `mapFeature.m`, que mapea los atributos en términos polinomiales de  $x_1$  y  $x_2$  hasta la sexta potencia ( $x_1, x_2, x_1^2, x_2^2, x_1*x_2, x_1*x_2^2, \dots$ ) y devuelve una matriz con 28 atributos, permitiendo de esta manera obtener una frontera de decisión mucho más compleja que se pueda adaptar mejor a los puntos. Modificar la regresión logística del apartado 2, para obtener un modelo a partir de los datos mapeados con 28 atributos. Una vez obtenido el modelo, haga una predicción de todo el conjunto de datos, imprimiendo por pantalla la tasa de acierto y la gráfica con la frontera de decisión.

**Notas:**

- 1) No se debe usar ninguna función de optimización avanzada sino que debe usar el método de descenso del gradiente implementado en el apartado 2).
- 2) Debe elegir alfa y el número de iteraciones de forma que el método de descenso del gradiente converja y al menos debe obtener una tasa de acierto mayor o igual que 82%.