Sesión 4

En este sesión, usaremos una de las tareas de clasificación de OpenML como ejemplo de examen. En particular, se utilizará la tarea *bank marketing* (data_id=1461). El objetivo de esta tarea es predecir cuándo un cliente de un banco firmará un depósito a plazo. Las características de entrada son numéricas (edad, balance en la cuenta, etc.) y nominales (trabajo, casado, educación etc.).

A continuación se muestra un resultado inicial (baseline) usando un clasificador de regresión logística estimado con los parámetros por defecto y dedicando un 90% de los datos para entrenamiento y un 10% para evaluación (random_state=23).

In [1]:
```
import warnings; warnings.filterwarnings("ignore"); import numpy as np
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

data_id = 1461
test_size = 0.1
X, y = fetch_openml(data_id=data_id, return_X_y=True, as_frame=False, parser="liac-arff")
# Valores de los parámetros por defecto: tol=1e-4, C=1e0, solver='lbfgs', max_iter=1e2
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=23)
clf = LogisticRegression(random_state=23).fit(X_train, y_train)
print(f'Test error: {(1 - accuracy_score(y_test, clf.predict(X_test)))*100:5.1f}%')
```

```
Test error:  10.9%
```

Ejercicio 1

Aplicando el clasificador de regresión logística con los valores de los parámetros por defecto excepto para el parámetro C, explora los valores del parámetro C en escala logarítmica para determinar su valor óptimo. Para cada valor explorado, muestra el error de clasificación en porcentaje sobre los conjuntos de entrenamiento y test. Usa random_state=23.

In [2]:
```
print('   solver     tol       C max_iter   etr   ete')
print('--------- ------- ------- -------- ----- -----')
for solver in ['lbfgs']:
    for tol in [1e-4]:
        for C in [1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3]:
            for max_iter in [100]:
                clf = LogisticRegression(solver=solver, tol=tol, C=C, max_iter=max_iter, random_state=23).fit(X_trai
                etr = 1 - accuracy_score(y_train, clf.predict(X_train))
                ete = 1 - accuracy_score(y_test, clf.predict(X_test))
                print(f'{solver:>9} {tol:.1e} {C:.1e} {max_iter:8d} {etr:5.1%} {ete:5.1%}')
```

```
   solver     tol       C max_iter   etr   ete
--------- ------- ------- -------- ----- -----
    lbfgs 1.0e-04 1.0e-03      100 11.3% 11.0%
    lbfgs 1.0e-04 1.0e-02      100 11.3% 10.8%
    lbfgs 1.0e-04 1.0e-01      100 11.3% 10.8%
    lbfgs 1.0e-04 1.0e+00      100 11.2% 10.9%
    lbfgs 1.0e-04 1.0e+01      100 11.3% 10.9%
    lbfgs 1.0e-04 1.0e+02      100 11.3% 10.8%
    lbfgs 1.0e-04 1.0e+03      100 11.3% 10.8%
```

Ejercicio 2

Aplicando el clasificador de regresión logística con los valores de los parámetros por defecto excepto para el parámetro C que debe ser fijado al mejor valor obtenido en el ejercicio 1, explora el máximo número de iteraciones en escala logarítimca para determinar su valor óptimo. Para cada valor explorado, muestra el error de clasificación en porcentaje sobre los conjuntos de entrenamiento y test. Usa random_state=23.

In [3]:
```
print('   solver     tol       C max_iter   etr   ete')
print('--------- ------- ------- -------- ----- -----')
for solver in ['lbfgs']:
    for tol in [1e-4]:
        for C in [1e1]:
            for max_iter in [100, 200, 500, 1000, 2000, 5000, 10000]:
                clf = LogisticRegression(solver=solver, tol=tol, C=C, max_iter=max_iter, random_state=23).fit(X_trai
                etr = 1 - accuracy_score(y_train, clf.predict(X_train))
                ete = 1 - accuracy_score(y_test, clf.predict(X_test))
                print(f'{solver:>9} {tol:.1e} {C:.1e} {max_iter:8d} {etr:5.1%} {ete:5.1%}')
```

```
   solver     tol       C max_iter   etr   ete
--------- ------- ------- -------- ----- -----
    lbfgs 1.0e-04 1.0e+01      100 11.3% 10.9%
    lbfgs 1.0e-04 1.0e+01      200 11.2% 10.5%
    lbfgs 1.0e-04 1.0e+01      500 11.1% 10.5%
    lbfgs 1.0e-04 1.0e+01     1000 11.2% 10.4%
    lbfgs 1.0e-04 1.0e+01     2000 11.1% 10.2%
    lbfgs 1.0e-04 1.0e+01     5000 10.9% 10.2%
    lbfgs 1.0e-04 1.0e+01    10000 11.0% 10.2%
```

Ejercicio 3

Aplicando el clasificador de regresión logística con los valores de los parámetros por defecto excepto para el parámetro C y el máximo número de iteraciones que deben ser fijados a los mejores valores obtenidos en los ejercicios 1 y 2, explora difertentes tipos de solver. Para cada solver explorado, muestra el error de clasificación en porcentaje sobre los conjuntos de entrenamiento y test. Usa random_state=23.

In [4]:
```
print('         solver     tol       C max_iter   etr   ete')
print('--------------- ------- ------- -------- ----- -----')
for solver in ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga']:
    for tol in [1e-4]:
        for C in [1e1]:
            for max_iter in [5000]:
                clf = LogisticRegression(solver=solver, tol=tol, C=C, random_state=23, max_iter=max_iter).fit(X_trai
                etr = 1 - accuracy_score(y_train, clf.predict(X_train))
                ete = 1 - accuracy_score(y_test, clf.predict(X_test))
                print(f'{solver:>15} {tol:.1e} {C:.1e} {max_iter:8d} {etr:5.1%} {ete:5.1%}')
```

```
         solver     tol       C max_iter   etr   ete
--------------- ------- ------- -------- ----- -----
          lbfgs 1.0e-04 1.0e+01     5000 10.9% 10.2%
      liblinear 1.0e-04 1.0e+01     5000 11.0% 10.3%
      newton-cg 1.0e-04 1.0e+01     5000 11.0% 10.2%
newton-cholesky 1.0e-04 1.0e+01     5000 11.0% 10.2%
            sag 1.0e-04 1.0e+01     5000 11.3% 10.8%
           saga 1.0e-04 1.0e+01     5000 11.3% 10.8%
```

Ejercicio 4

De acuerdo a los resultados obtenidos, se podría afirmar que esta tarea es linealmente separable? Razona la respuesta.

No se podría afirmar que la tarea sea linealmente separable ya que los porcentajes de error obtenidos están lejos de cero por lo que es razonable afirmar que esta tarea no debe ser linealmente separable.