Sessió 4

En aquest sessió, usarem una de les tasques de classificació d'OpenML com a exemple d'examen. En particular, s'utilitzarà la tasca *bank màrqueting* (data_id=1461). L'objectiu d'aquesta tasca és predir quan un client d'un banc signarà un dipòsit a termini. Les característiques d'entrada són numèriques (edat, balanç en el compte, etc.) i nominals (treball, casat, educació etc.).

A continuació es mostra un resultat inicial (baseline) usant un classificador de regressió logística estimat amb els paràmetres per defecte i dedicant un 90% de les dades per a entrenament i un 10% per a avaluació (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")
# Valors dels paràmetres per defecte: 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%
```

Exercici 1

Aplicant el classificador de regressió logística amb els valors dels paràmetres per defecte excepte per al paràmetre C, explora els valors del paràmetre C en escala logarítmica per a determinar el seu valor òptim. Per a cada valor explorat, mostra l'error de classificació en percentatge sobre els conjunts d'entrenament i 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%
```

Exercici 2

Aplicant el classificador de regressió logística amb els valors dels paràmetres per defecte excepte per al paràmetre C que ha de ser fixat al millor valor obtingut en l'exercici 1, explora el màxim nombre d'iteracions en escala logarítimca per a determinar el seu valor òptim. Per a cada valor explorat, mostra l'error de classificació en percentatge sobre els conjunts d'entrenament i 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%
```

Exercici 3

Aplicant el classificador de regressió logística amb els valors dels paràmetres per defecte excepte per al paràmetre C i el màxim nombre d'iteracions que han de ser fixats als millors valors obtinguts en els exercicis 1 i 2, explora difertentes tipus de solver. Per a cada solver explorat, mostra l'error de classificació en percentatge sobre els conjunts d'entrenament i 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%
```

Exercici 4

D'acord amb els resultats obtinguts, es podria afirmar que aquesta tasca és linealment separable? Raona la resposta.

No es podria afirmar que la tasca siga linealment separable ja que els percentatges d'error obtinguts estan lluny de zero per la qual cosa és raonable afirmar que aquesta tasca no ha de ser linealment separable.