Анализ данных

Хашин С.И.
http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html

Ивановский университет

.

Рекламная кампания банка

.

Иваново-2023

План

Рекламная кампания банка

Банк собирается проводить рекламную кампанию среди 30 000 своих клиентов. Она хоть как-то заинтересует 10% клиентов, то есть 3000. Для каждого клиента имеется вектор из 50 чисел, его характеризующий (признаки).
Написать программу, которая по признакам определяет заинтересует ли реклама клиента.
Качество программы характеризуется двумя типами ошибок:

- Количество клиентов, которым эта реклама не интересна, но попавшие в выборку,
- Количество клиентов, которым реклама интересна, но не попавшие в выборку.

Качество работы алгоритма будет оцениваться при помощи показателя AUC (area under curve) — площади под ROC-кривой.
Проще: как количество ошибок 1-го типа плюс количество ошибок 2-го типа, умноженное на 10.

Описание данных

Данные содержат записи о 15 223 клиентов,
классифицированных на два класса:

- 1 — отклик был (1812 клиентов),
- 0 — отклика не было (13411 клиентов).

Данные и описания полей лежат в архиве
`bank_contest2011.7z`.
Objects.csv — данные, обучающая матрица
Target.csv — целевая функция (0 или 1) для каждого клиента.
PropertyDesciption.xlsx — описание полей в Objects.csv

Objects.csv

```
49,0,0,0,2,1,0,0,0,0,0,0,0,3,5000,0,0,0,0,0,1,1,1,1,...
32,0,0,0,3,3,1,0,0,0,1,0,1,3,12000,1,1,1,1,1,1,1,1,...
26,0,0,1,0,0,0,3,4,5,2,0,1,3,18000,10,10,10,10,5,1,...
62,1,1,1,3,0,1,1,7,0,0,0,1,3,7000,11,11,11,11,3,1,1,...
45,0,0,0,3,1,0,2,3,0,2,0,4,3,12000,9,9,9,9,4,1,1,1,...
38,0,0,0,0,0,1,3,0,3,6,2,0,1,4,20000,12,12,12,12,5,1,...
41,0,0,0,0,0,0,3,2,0,7,1,0,1,5,60000,10,10,10,10,5,1,...
65,1,1,0,2,0,0,0,NaN,NaN,NaN,NaN,NaN,4,10000,13,13,...
29,0,0,0,1,1,0,0,8,0,2,0,1,3,5200,14,14,14,14,6,1,1,...
34,0,0,1,0,0,1,1,9,0,0,0,1,5,35000,15,15,15,15,5,0,...
...
```

NaN — Not A Number

На Питоне

```
def read_data():    # return Y,X
    X = np.loadtxt('Objects.csv', delimiter=',')
    Y = np.loadtxt('Target.csv').astype(int)
    return Y, X

def clear0(X):    # нулевая очистка матрицы X, !на месте!
    X[np.isnan(X)] = 0

def binarize(X, k): # бинаризация k-го столбца матрицы
    X0 = X[:, :k]
    X1 = X[:, k].astype(int)
    maxX1 = X1.max()
    X1bin = np.eye(maxX1+1)[X1]
    X2 = X[:, k:]
    return np.hstack((X0, X1bin, X2))
```

Бинаризация и очистка

```
Y, X = read_data()
clear0(X)
 X = binarize(X, 19)      # 20, Регион РФ
 X = binarize(X, 18)      # 19, Область торговой точки где бы
 X = binarize(X, 17)      # 18, Почтовый адрес области
 X = binarize(X, 16)      # 17, Область фактического пребыван
 X = binarize(X, 15)      # 16, Область регистрации
 X = binarize(X, 11)      # 12, Направление деятельности внут
 X = binarize(X, 10)      # 11, Форма собственности компании
 X = binarize(X, 9)       # 10, Должность
 X = binarize(X, 8)       # 9, Отрасль работы
 X = binarize(X, 7)       # 8, Семейное положение
 X = X[:,~(X == 0).all(0)] # удаление нулевых столбцов
 print('после удаления нулей:', X.shape)
```

Бинаризация и очистка

```
np.savetxt('t:\\0.csv', X, fmt='%9.0f', delimiter=',',
           header='x,y', comments='')
print('после бинаризации:', X.shape)
X = X[:,~(X == 0).all(0)] # удаление нулевых столбцов
print('после удаления нулей:', X.shape)
np.savetxt('t:\\1.csv', X, fmt='%9.0f', delimiter=',',
           header='x,y', comments='')
```

На Питоне

```
def lin_reg1(A,b):
    '''
    Линейная регрессия: b ~= w[0] + A.dot(w[1:])
    :param A: обучающие вектора
    :param b: значения функции на каждом вектора
    :return: w - вектор
    '''
    # добавление столбца из единиц в начало матрицы
    A = np.hstack((np.ones(len(A)).reshape((-1,1)), A))
    AtA = A.T.dot(A)
    eps = 1e-8*abs(AtA).max()
    AtA += eps*np.eye(len(AtA))
    Atb = A.T.dot(b.reshape(-1))
    return np.linalg.solve(AtA,Atb)
```

На Питоне

```
def TP_FP_FN_TN(Y, X, w, border):
    TP = FP = FN = TN = 0
    for i,x1 in enumerate(X):
        a_x = w[0] + np.dot(x1, w[1:])
        if y[i]>0:
            if a_x >= border: TP += 1
            else:             FN += 1
        else:
            if a_x >= border: FP += 1
            else:             TN += 1
    return TP, FP, FN, TN
```

На Питоне, ROC-кривая

Напомню, что X, Y, w уже готовы

```
cx, cy = [], []
for ib in range(-150, 150):
    TP, FP, FN, TN = TP_FP_FN_TN(Y, X, w, ib/100)
    cx.append(FP/(FP+TN))
    cy.append(TP/(TP+FN))
plt.scatter(cx, cy, s=1, color='red')
plt.show()
```

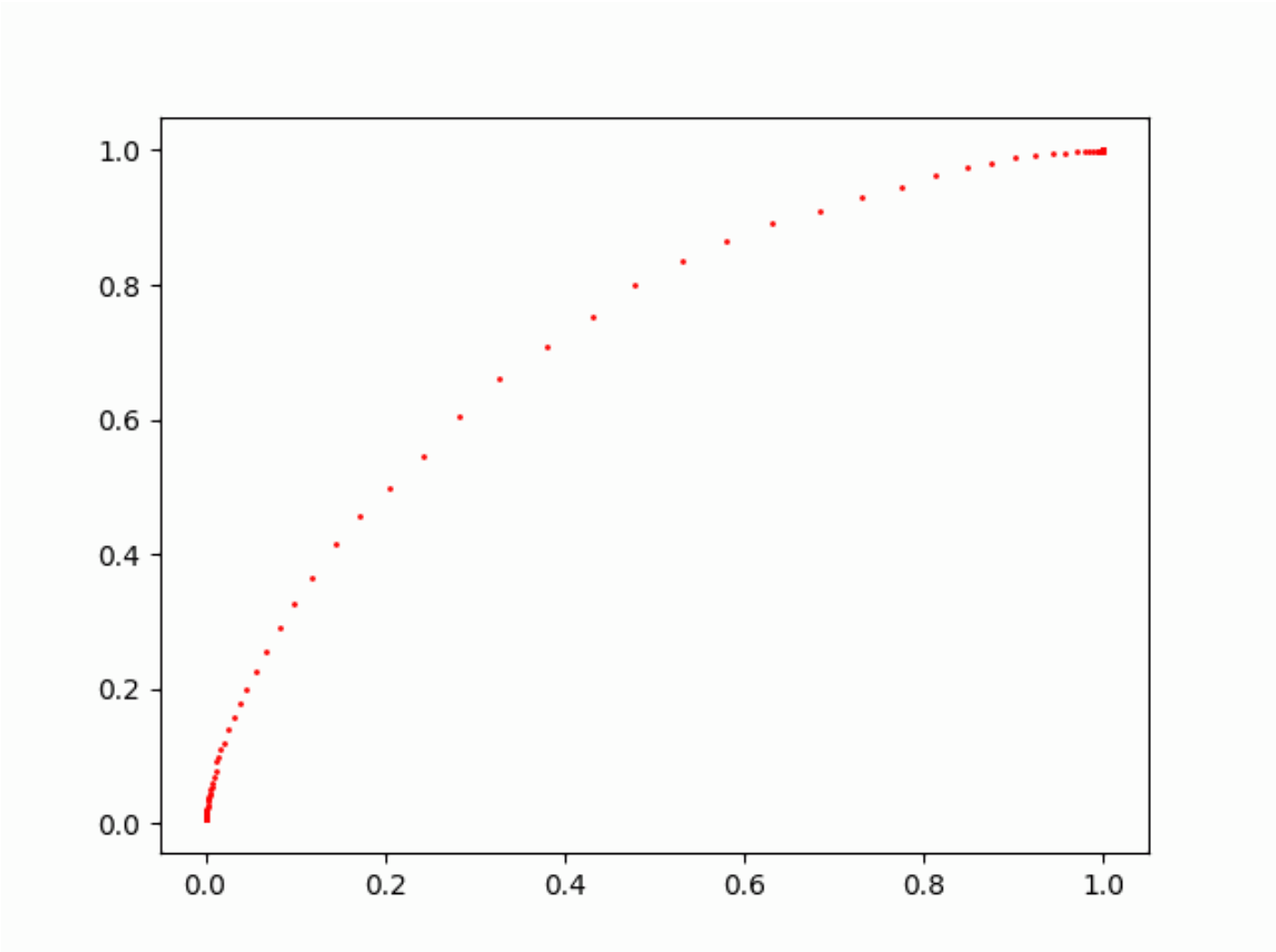
На Питоне

```
def trianlge_area(x0,y0,x1,y1,x2,y2):#площадь треугольника
    x1 -= x0;  y1 -= y0
    x2 -= x0;  y2 -= y0
    return (x1*y2 - x2*y1)/2

def AUC(x, y): #
    area = 0
    for i in range(1, len(x)):
        area+=trianlge_area(1,0,x[i-1],y[i-1],x[i],y[i])
    return area
```

На Питоне

```
w = lin_reg(Y,X)
cx, cy = [], []
for ib in range(-50, 100):
    TP, FP, FN, TN = TP_FP_FN_TN(X, Y, w, ib/100)
    cx.append(FP/(FP+TN))
    cy.append(TP/(TP+FN))
plt.scatter(cx, cy, s=1, color='red') #plt.show()
plt.savefig('ROC_bank.png', dpi = 100) # save as png

print(f'AUC={AUC(cx, cy):7.3f}')
> AUC=  0.725
```

ROC-кривая



Задание

1. Проанализируйте, какие показатели больше влияют не результат. Не забудьте, что при бинаризации номера показателей изменяются.
Учтите, что большинство показателей лежит в пределах от 0 до 1, а, например, зарплата, в пределах от 2000 до 200 тыс..
Поэтому показатели стоит нормализовать.
2. Рассмотрим целевую функцию S(border) равную:
10*количество клиентов, кому интересна реклама, но её не дали
+количество клиентов, кому интересна не реклама, но её дали
При каком параметре border S(border) будет минимальна?
Напечатайте матрицу ошибок в этом случае.