مشروع نهاية الفصل 3

مقرر Machine Learning — Classification — مشروع التصنيف

الهدف: تبني مصنّف أرقام، وتقيّمه بـ confusion matrix وتعمل error analysis: أنهي رقمين بيتلخبطوا أكتر؟

الداتا: sklearn.datasets.load_digits (١٧٩٧ صورة ٨×٨، ١٠ أرقام). مصدرها UCI. مدمجة في scikit-learn.

الأسئلة العملية

1 الداتا (Look at the data)

- حمّل digits واطبع عدد الصور وعدد كل رقم.

2 قسّم (Split)

- قسّم ٨٠/٢٠ بـ stratify و random_state=0.

3 Baseline (Baseline)

- دقة DummyClassifier(strategy='most_frequent').

4 درّب (Train)

- درّب Pipeline (StandardScaler + LogisticRegression(max_iter=2000)) واطبع دقة الـ test.

5 Error analysis (Error analysis)

- اطبع الـ confusion matrix.
- لاقي أكتر زوج (حقيقي، متوقّع) بيتلخبط (من غير القطر).

6 Precision و Recall (Precision and recall)

- اطبع precision و recall لرقم ٨ بعد تحويله لمشكلة (8 ولا مش 8).

بونس

- بونس: قارن مع SVC(kernel='rbf').
- بونس: جرّب cross_val_predict وارسم confusion matrix على الـ training.

أسئلة تفكير (اكتب إجابة في سطرين)

1. ليه الـ accuracy لوحدها ممكن تضلّلنا لو فئة نادرة؟
2. إمتى نفضّل recall على precision؟

ماذا تسلّم؟

- ملف الكود (أو notebook) بيشتغل من الأول للآخر.
- فقرة قصيرة (٥ أسطر) بتلخّص النتيجة وإيه اللي اتعلمته.

الحل النموذجي

جرّب بنفسك الأول، وبعدين قارن. الأرقام الحقيقية في النتيجة تحت.

الكود

```
# dataset ships inside scikit-learn (no download needed)
import numpy as np
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.dummy import DummyClassifier
from sklearn.metrics import confusion_matrix, precision_score, recall_score

X, y = load_digits(return_X_y=True)
print('images:', len(y), '| per digit:', np.bincount(y).tolist())
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.2, random_state=0,
stratify=y)
print('baseline:', round(DummyClassifier(strategy='most_frequent').fit(X_tr,
y_tr).score(X_te, y_te), 3))
m = make_pipeline(StandardScaler(), LogisticRegression(max_iter=2000)).fit(X_tr, y_tr)
print('test accuracy:', round(m.score(X_te, y_te), 3))
cm = confusion_matrix(y_te, m.predict(X_te))
print(cm)
off = cm.copy(); np.fill_diagonal(off, 0)
i, j = np.unravel_index(off.argmax(), off.shape)
print('most confused: true', i, 'predicted as', j, '->', int(off[i, j]), 'times')
p = m.predict(X_te)
print('digit 8 precision:', round(precision_score(y_te == 8, p == 8), 3), 'recall:',
round(recall_score(y_te == 8, p == 8), 3))
```

النتيجة

```
images: 1797 | per digit: [178, 182, 177, 183, 181, 182, 181, 179, 174, 180]
baseline: 0.1
test accuracy: 0.967
[[36  0  0  0  0  0  0  0  0  0]
 [ 0 35  0  0  0  0  0  0  1  0]
 [ 0  1 34  0  0  0  0  0  0  0]
 [ 0  0  0 37  0  0  0  0  0  0]
 [ 0  1  0  0 34  0  0  0  1  0]
 [ 0  0  0  0  0 36  0  1  0  0]
 [ 0  1  0  0  0  0 35  0  0  0]
 [ 0  0  0  0  0  0  0 36  0  0]
 [ 0  4  0  0  0  0  0  0 30  1]
 [ 0  0  0  0  0  1  0  0  0 35]]
```

```
most confused: true 8 predicted as 1 -> 4 times
digit 8 precision: 0.938 recall: 0.857
```

ملاحظات على الإجابات

- الداتا متوازنة تقريباً فالـ baseline حوالي ١٠٪.
- الـ confusion matrix بيوريك الأخطاء الحقيقية، مش بس رقم واحد للدقة.
- الـ precision بتجاوب: من اللي قلنا عليه ٨، كام فعلاً ٨؟ والـ recall: من كل الـ ٨ الحقيقية، كام لقينا؟
- نفضّل recall لما تفويت الحالة أخطر من إنذار كاذب، زي الفحص الطبي.

Credits: topic follows the matching chapter of Aurélien Géron, Hands-On Machine Learning (3rd ed.). Questions, code and explanations are original.