# مشروع نهاية الفصل ١

مقرر Machine Learning — The Machine Learning Landscape

**الهدف:** تطبّق على داتا حقيقية صغيرة (نبيذ من ٣ أنواع) كل اللي اتعلمته في الفصل: نوع المشكلة، الـ baseline، الـ validation، الـ test set، والفرق بين Supervised وUnsupervised.

**الوقت المتوقع:** ساعة تقريباً. **الأدوات:** Python مع scikit-learn، وتقدر تشغّل الكود كله في Google Colab المجاني.

**الداتا:** sklearn.datasets.load_wine (١٧٨ عينة، ١٣ خاصية كيميائية، ٣ أنواع). مدمجة في scikit-learn ومش محتاجة تحميل.

## الأسئلة العملية

### 1 افهم المشكلة (بالكلمات) (Understand the problem)

- إيه نوع المشكلة هنا: Supervised ولا Unsupervised؟ Classification ولا Regression؟ ليه؟
- مين الـ features ومين الـ label في داتا النبيذ؟
- اكتب جملة واحدة تشرح ليه مش هتكتب قواعد بإيدك لحل المشكلة دي.

### 2 الداتا (Look at the data)

- حمّل load_wine واطبع شكل الداتا (rows, features) وعدد الأمثلة في كل فئة.
- هل الفئات متوازنة تقريباً؟ إيه تأثير ده على اختيار مقياس النجاح؟

### 3 قسّم الداتا (Split the data)

- قسّم الداتا 70/30 بـ stratify=y و random_state=42، واحتفظ بالـ test set على جنب. ممنوع تستخدمه قبل الخطوة ٥.

### 4 الـ Baseline (Baseline)

- درّب DummyClassifier(strategy='most_frequent') واطبع دقته على الـ test. ده الرقم اللي لازم نتفوّق عليه.

### 5 اختر k (Choose k)

- جرّب KNeighborsClassifier بـ k = 1, 5, 15, 40 باستخدام cross_val_score (cv=5) على الـ training set بس.
- اختار أحسن k. لو نتيجتين متقاربتين، اكتب ملاحظتك.

### 6 الاختبار النهائي (Final test)

- درّب النموذج بالـ k اللي اخترته على كل الـ training set، واطبع دقته على الـ test مرة واحدة بس.

## بونس

- بونس ١: حط StandardScaler قبل الـ kNN في pipeline وقارن الدقة. إيه اللي اتغيّر؟ فكّر ليه (ستة أرقام بمقاييس مختلفة جداً).
- بونس ٢: من غير ما تستخدم الـ labels، جرّب KMeans(n_clusters=3) على الداتا بعد scaling، وقارن المجموعات بالفئات الحقيقية بـ adjusted_rand_score. ده Supervised ولا Unsupervised؟

## أسئلة تفكير (اكتب إجابة في سطرين)

1. ليه ماينفعش نختار k بالنظر لنتيجة الـ test set؟
2. لو دقة الـ training عالية جداً والـ test واطية، ده معناه إيه؟ وإيه ٢ حلول ممكنة؟
3. إمتى بنفضّل Batch ومتى Online في مشروع زي ده لو الداتا بتزيد كل يوم؟

## ماذا تسلّم؟

- ملف الكود (أو notebook) بيشتغل من الأول للآخر.
- فقرة قصيرة (٥ أسطر) بتلخّص: إيه النتيجة، وإيه اللي اتعلمته، وإيه اللي كنت هتجرّبه بعدين.

# الحل النموذجي

جرّب بنفسك الأول، وبعدين قارن.

## الكود

```
import numpy as np
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.dummy import DummyClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score

# Task 1: look at the data
X, y = load_wine(return_X_y=True)
print("rows, features:", X.shape)
print("examples per class:", np.bincount(y).tolist())

# Task 2: split once, keep the test set aside
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.3, random_state=42,
stratify=y)

# Task 3: baseline
dummy = DummyClassifier(strategy="most_frequent").fit(X_tr, y_tr)
print("baseline accuracy:", round(dummy.score(X_te, y_te), 3))

# Task 4: choose k with 5-fold cross-validation on the training set only
best_k, best = None, 0
for k in (1, 5, 15, 40):
    s = cross_val_score(KNeighborsClassifier(n_neighbors=k), X_tr, y_tr, cv=5).mean()
    print("k =", k, "-> validation accuracy:", round(s, 3))
    if s > best:
        best_k, best = k, s
print("chosen k:", best_k)

# Task 5: judge the chosen model once on the test set
final = KNeighborsClassifier(n_neighbors=best_k).fit(X_tr, y_tr)
print("test accuracy:", round(final.score(X_te, y_te), 3))

# Bonus A: put the features on the same scale
scaled = make_pipeline(StandardScaler(),
KNeighborsClassifier(n_neighbors=best_k)).fit(X_tr, y_tr)
print("test accuracy with scaling:", round(scaled.score(X_te, y_te), 3))

# Bonus B: unsupervised, ignore the labels and compare the groups to them
Xs = StandardScaler().fit_transform(X)
groups = KMeans(n_clusters=3, n_init=10, random_state=42).fit_predict(Xs)
```

```
print("agreement of KMeans groups with real classes (ARI):",
round(adjusted_rand_score(y, groups), 3))
```

## النتيجة

```
rows, features: (178, 13)
examples per class: [59, 71, 48]
baseline accuracy: 0.389
k = 1 -> validation accuracy: 0.718
k = 5 -> validation accuracy: 0.71
k = 15 -> validation accuracy: 0.702
k = 40 -> validation accuracy: 0.718
chosen k: 40
test accuracy: 0.704
test accuracy with scaling: 0.944
agreement of KMeans groups with real classes (ARI): 0.897
```

## إجابات الأسئلة

- **نوع المشكلة:** Supervised classification، لأن عندنا نوع النبيذ الصحيح لكل عينة (label) والهدف فئة من ٣ فئات.
- **Features و label:** الـ features هي الـ ١٣ قياس كيميائي؛ الـ label هو نوع النبيذ (٠ أو ١ أو ٢).
- **ليه مش قواعد بإيدنا؟** الفروق بين الأنواع موزّعة على ١٣ قياس بشكل معقّد، والنموذج بيلقطها من الأمثلة أسهل من كتابة قواعد.
- **توازن الفئات:** ٥٩ / ٧١ / ٤٨، متوازنة تقريباً، فالـ accuracy مقياس معقول هنا. الـ baseline حوالي ٠.٣٩.
- **اختيار k:** النتايج كلها قريبة (٠.٧٠ إلى ٠.٧٢) فالفرق ضئيل ومش مقنع. اخترنا k=40 لأنه أعلى رقم بفارق بسيط جداً، وده تذكير إن الفروق الصغيرة في الـ validation ممكن تكون ضوضاء.
- **الـ test:** ٠.٧٠ تقريباً، أحسن من الـ baseline بوضوح لكن مش كويس كفاية.
- **بونس ١:** مع الـ scaling الدقة وصلت ٠.٩٤، لأن الـ kNN بيعتمد على المسافات وبعض الـ features أرقامها كبيرة جداً وبتطغى على الباقي. ده مثال إن تجهيز الداتا ممكن يفرق أكتر من اختيار النموذج (هنشوفه في الفصل ٢).
- **بونس ٢:** Unsupervised، لأننا ما استخدمناش الـ labels في التدريب. الـ ARI ≈ ٠.٩٠ يعني المجموعات قريبة جداً من الأنواع الحقيقية.
- **ليه مش بنختار k من الـ test؟** لأن ده بيسرّب معلومات من الـ test للقرار، فالنتيجة النهائية بتبقى متفائلة زيادة ومش بتعكس الأداء على داتا جديدة.

- **Training عالية وtest واطية:** ده overfitting. حلول: نموذج أبسط أو regularization، أو داتا أكتر/أنضف.
- **Batch أم Online؟** لو الداتا بتزيد كل يوم ومحتاجين نموذج يتحدّث بسرعة أو الداتا كبيرة جداً: Online. لو بنعيد التدريب كل فترة والداتا بتدخل الذاكرة: Batch كافي.

Credits: topic follows Chapter 1 of Aurélien Géron, Hands-On Machine Learning (3rd ed.). Questions, code and explanations are original.