# مشروع نهاية الفصل 7

مقرر Machine Learning — Ensemble Learning and Random Forests — مشروع الـ Ensembles

**الهدف:** تقارن شجرة واحدة بـ Random Forest و Gradient Boosting و Voting بالـ cross-validation.

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

## الأسئلة العملية

### 1 جهّز (Prepare)

- حمّل digits وقسّمها ٨٠/٢٠ بـ stratify و random_state=0.

### 2 النماذج (Models)

- جهّز شجرة واحدة، RandomForest (200 شجرة)، GradientBoosting (100 شجرة).

### 3 قارن (Compare)

- قارنهم بـ 3-fold cross-validation على الـ training.

### 4 Voting (Voting)

- اعمل VotingClassifier(voting='soft') من الـ forest و LogisticRegression واطبع الدقة.

### 5 النهائي (Final)

- اختار أحسن واحد وقيسه على الـ test.

## بونس

- بونس: استخدم oob_score في الـ forest.
- بونس: اطبع أهم ١٠ بكسلات.

## أسئلة تفكير (اكتب إجابة في سطرين)

1. ليه المجموعة بتتفوق على الشجرة الواحدة؟
2. إيه عيب الـ boosting مقارنة بالـ bagging؟

## 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)
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier,
VotingClassifier
from sklearn.linear_model import LogisticRegression

X, y = load_digits(return_X_y=True)
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.2, random_state=0,
stratify=y)
models = {'tree': DecisionTreeClassifier(random_state=0),
 'forest': RandomForestClassifier(n_estimators=200, random_state=0),
 'boosting': GradientBoostingClassifier(n_estimators=100, max_depth=3, random_state=0)}
cv = {n: cross_val_score(m, X_tr, y_tr, cv=3).mean() for n, m in models.items()}
for n, s in cv.items():
    print(n.ljust(9), 'cv accuracy', round(s, 3))
vote = VotingClassifier([('rf', models['forest']), ('lr',
LogisticRegression(max_iter=3000))], voting='soft')
print('soft voting cv:', round(cross_val_score(vote, X_tr, y_tr, cv=3).mean(), 3))
best = max(cv, key=cv.get)
print('best single family:', best, '| test', round(models[best].fit(X_tr,
y_tr).score(X_te, y_te), 3))
```

## النتيجة

```
tree      cv accuracy 0.825
forest    cv accuracy 0.969
boosting  cv accuracy 0.962
soft voting cv: 0.969
best single family: forest | test 0.978
```

## ملاحظات على الإجابات

- المجموعة بتقلّل التباين لأن أخطاء الأشجار المختلفة بتتعادل.
- الـ boosting بيدرّب بالتتابع فمش سهل يتوازى، وممكن يعمل overfitting لو الجولات كتير.
- اختار الأحسن بالـ cross-validation مش بالـ test.

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