

PRACTICE CLASS 2

Practice: evaluation clinic

Find the mistakes in four experiments, then fix them

Today's key line of code

```
cross_val_score(pipeline, X_tr, y_tr, cv=5)
```

How today works

1

Part A: by hand (20 min)

Metrics from a confusion matrix, choosing a depth, choosing a metric.

2

Part B: be the reviewer (30 min)

Four short experiments. Each has one mistake. Name it before running any code.

3

Part C: fix it (40 min)

Rewrite each experiment correctly and compare the numbers.

Part A: by hand

1 A1: metrics

TP = 30, FP = 10, FN = 5, TN = 55.
Compute accuracy, precision, recall and F1.

2 A2: choose a depth

Depth 1: train 0.80, val 0.78. Depth 3: 0.92, 0.90. Depth 6: 0.99, 0.88. Depth 12: 1.00, 0.85. Which depth, and why?

3 A3: choose a metric

Which metric matters most for (a) a cancer screening test, (b) a spam filter, (c) a balanced digit recogniser?

Experiment 1: what is wrong?

A student reports: 'My decision tree is 100% accurate on the breast cancer data!'

Code

```
from sklearn.datasets import load_breast_cancer
from sklearn.tree import DecisionTreeClassifier
X, y = load_breast_cancer(return_X_y=True)
tree = DecisionTreeClassifier(random_state=0)
tree.fit(X, y)
print(tree.score(X, y))
```

Output

1.0

Experiment 2: what is wrong?

A student scales the data, then splits it, and reports the test accuracy.

Code

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
X_scaled = StandardScaler().fit_transform(X)
X_tr, X_te, y_tr, y_te = train_test_split(
    X_scaled, y, test_size=0.3, random_state=0, stratify=y)
knn = KNeighborsClassifier().fit(X_tr, y_tr)
print(knn.score(X_te, y_te))
```

Output

```
0.9473684210526315
```

Experiment 3: what is wrong?

A student tries many k values and reports the best test accuracy as the final result.

Code

```
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=0.3, random_state=0, stratify=y)
best = 0
for k in range(1, 40):
    acc = KNeighborsClassifier(n_neighbors=k).fit(
        X_tr, y_tr).score(X_te, y_te)
    best = max(best, acc)
print(best)
```

Output

```
0.9239766081871345
```

Experiment 4: what is wrong?

A fraud detector on 1,000 transactions (2% fraud) reports 98% accuracy. The data is invented.

Code

```
import numpy as np
from sklearn.dummy import DummyClassifier
rng = np.random.default_rng(0)
Xf = rng.normal(size=(1000, 5))
yf = np.array([1] * 20 + [0] * 980)
model = DummyClassifier(strategy="most_frequent").fit(Xf, yf)
print(model.score(Xf, yf))
```

Output

0.98

Part C: fix the experiments

1 Fix 1

Split first. Report training and test accuracy of the tree.

2 Fix 2

Put StandardScaler and k-NN in a pipeline, fitted on training data only.

3 Fix 3

Choose k with 5-fold cross-validation on the training data; test once.

4 Fix 4

Report recall and precision for the fraud class. What do they show?

5 Fix 5: all together

One honest experiment on breast cancer: pipeline, CV to choose k, one test score, `classification_report`.

Solutions

Show after students have reviewed and fixed the experiments.

Part A solutions

Task	Answer
A1	accuracy $85/100 = 0.85$ · precision $30/40 = 0.75$ · recall $30/35 = 0.857$ · F1 = $2(0.75)(0.857)/(1.607) = 0.80$
A2	Depth 3: best validation score (0.90). Depth 12 has perfect training accuracy but is overfitting.
A3	(a) recall: missing a sick patient is worst · (b) precision: losing real messages is worst · (c) accuracy is fine when classes are balanced

Fix 1: split before scoring

Training accuracy: 1.0, memorised · Test accuracy: the honest number · Report the test number · The gap reveals overfitting

Code

```
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=0.3, random_state=0, stratify=y)
tree = DecisionTreeClassifier(random_state=0).fit(X_tr, y_tr)
print(tree.score(X_tr, y_tr))
print(round(tree.score(X_te, y_te), 3))
```

Output

```
1.0
0.906
```

Fix 2: scale inside a pipeline

The scaler is fitted on training rows only • Here the number barely changes • On other data leakage can inflate scores a lot

- The method is wrong even when the number looks similar

Code

```
from sklearn.pipeline import make_pipeline
pipe = make_pipeline(StandardScaler(), KNeighborsClassifier())
pipe.fit(X_tr, y_tr)
print(round(pipe.score(X_te, y_te), 3))
```

Output

0.947

Fix 3: choose k with cross-validation

Every k is judged on validation folds · The test set is untouched until the end · One final, honest test score · Here it is higher than Experiment 3, because we also scaled

Code

```
from sklearn.model_selection import cross_val_score
cv_means = {}
for k in [1, 3, 5, 9, 15, 25]:
    p = make_pipeline(StandardScaler(),
                      KNeighborsClassifier(n_neighbors=k))
    cv_means[k] = cross_val_score(p, X_tr, y_tr, cv=5).mean()
best_k = max(cv_means, key=cv_means.get)
final = make_pipeline(StandardScaler(),
                      KNeighborsClassifier(n_neighbors=best_k))
final.fit(X_tr, y_tr)
print(best_k, round(final.score(X_te, y_te), 3))
```

Output

```
5 0.947
```

Fix 4: look at the rare class

- Accuracy 0.98, as reported
- Recall for fraud: 0, it catches nothing
- Precision is undefined: it never says fraud
- A useless model hiding behind accuracy

Code

```
from sklearn.metrics import recall_score, precision_score
pred = model.predict(Xf)
print(model.score(Xf, yf))
print(recall_score(yf, pred))
print(precision_score(yf, pred, zero_division=0))
```

Output

```
0.98
0.0
0.0
```

Fix 5: one honest experiment, start to finish

- Split, pipeline, cross-validation, one test
- Report per-class precision and recall
- malignant = class 0 in this dataset
- Use this structure in your project

Code

```
from sklearn.metrics import classification_report
pred = final.predict(X_te)
print(classification_report(y_te, pred, digits=3,
                           target_names=["malignant", "benign"]))
```

Output

	precision	recall	f1-score	support
malignant	0.966	0.891	0.927	64
benign	0.938	0.981	0.959	107
accuracy			0.947	171
macro avg	0.952	0.936	0.943	171
weighted avg	0.948	0.947	0.947	171

Homework P2

Due before Lecture 13 • [date]

1 Finish

Complete all five fixes in your Practice2 notebook, with a text cell explaining each mistake in one sentence.

2 Review a friend

Swap notebooks with a classmate and look for any remaining evaluation mistake. Write one comment.

3 Your project

Which evaluation mistake is most likely in a project you might do? How will you avoid it?

Submit on [LMS / Google Classroom]: your notebook file HWP2_<your ID>.ipynb, or a Colab link set to “Anyone with the link can view”.