

LECTURE 8

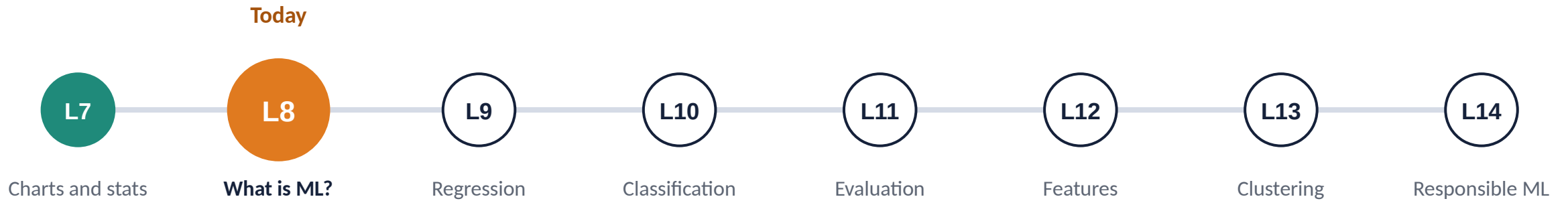
What is machine learning?

Learning from examples: the ideas behind every model

Today's key line of code

```
model.fit(X_train, y_train)  
model.predict(X_test)
```

The road ahead: machine learning



Today you will

- 1 Explain machine learning as learning a function from examples
- 2 Tell supervised from unsupervised learning
- 3 Tell classification from regression
- 4 Use the fit / predict pattern with any scikit-learn model
- 5 Explain why we judge a model on data it has never seen

Machine learning in one sentence

1

Examples in

We give the computer many examples: inputs (features) and, often, the right answers (labels).

2

Pattern learned

An algorithm finds a rule, called a model, that maps inputs to answers.

3

New answers out

The model predicts answers for inputs it has never seen. That is the whole point.

Supervised or unsupervised?

Supervised (labels given)

- Every example has the right answer
- Goal: predict the answer for new inputs
- Spam filter, price prediction, diagnosis
- Lectures 9 to 12

Unsupervised (no labels)

- Only inputs, no answers
- Goal: find structure, such as groups
- Customer segments, topic grouping
- Lecture 13

Classification or regression?

Ask: is the answer a category or a number?

Question	Answer looks like	Type
Is this message spam?	spam / ham	Classification
What mark will this student get?	a number, e.g. 78.5	Regression
Which flower species is this?	one of 3 names	Classification
How many units will sell next week?	a number	Regression
Will this loan be repaid?	yes / no	Classification

The machine learning workflow



The same workflow as Lecture 6, for any dataset. Most of the work is before training.

Meet a classic dataset: iris flowers

150 flowers, 4 measurements each (in cm), 3 species. It ships with scikit-learn, so it works offline.

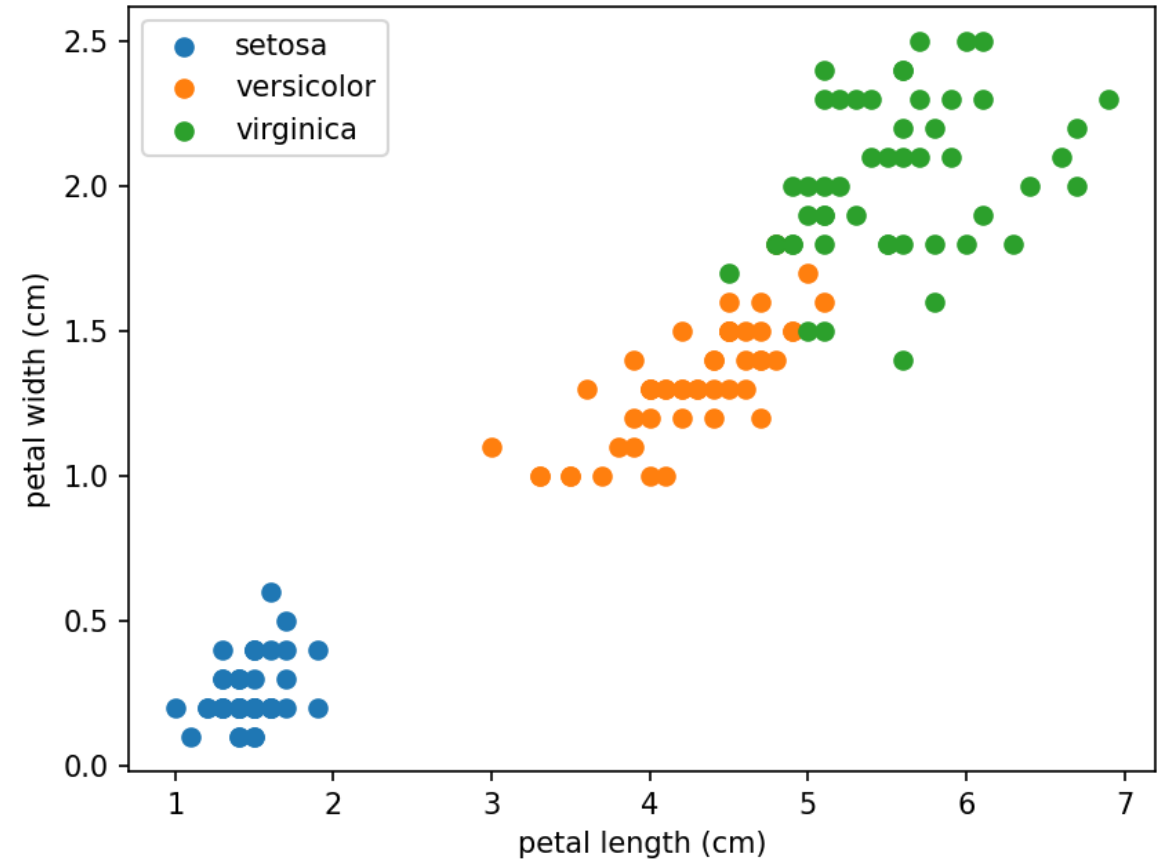
sepal length	sepal width	petal length	petal width	species (label)
5.1	3.5	1.4	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.3	3.3	6.0	2.5	virginica
4.9	3.0	1.4	0.2	setosa
6.4	3.2	4.5	1.5	versicolor

Look first: can the classes be told apart?

Two petal measurements already separate the species well. Machine learning finds such patterns automatically, in many dimensions.

Code

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
iris = load_iris(as_frame=True)
X = iris.data
y = iris.target
for k, name in enumerate(iris.target_names):
    part = X[y == k]
    plt.scatter(part["petal length (cm)"],
                part["petal width (cm)"], label=name)
plt.xlabel("petal length (cm)")
plt.ylabel("petal width (cm)")
plt.legend()
plt.show()
```



The same pattern for every model

- Split first, with the same function as Lecture 6
- **fit** learns from training data
- **score** reports accuracy on test data
- Swap the model; the code stays the same

Code

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y)
for model in [KNeighborsClassifier(),
               DecisionTreeClassifier(random_state=0),
               GaussianNB()]:
    model.fit(X_tr, y_tr)
    acc = model.score(X_te, y_te)
    print(type(model).__name__, round(acc, 3))
```

Output

```
KNeighborsClassifier 0.978
DecisionTreeClassifier 0.978
GaussianNB 0.911
```

100% on training data: a perfect model?

Predict first

Code

```
one_nn = KNeighborsClassifier(n_neighbors=1)
one_nn.fit(X_tr, y_tr)
print(one_nn.score(X_tr, y_tr))
print(round(one_nn.score(X_te, y_te), 3))
```



A 1-nearest-neighbour model scores 100% on its training data. What will it score on the test data?

Answer: 100% on training data: a perfect model?

Code

```
one_nn = KNeighborsClassifier(n_neighbors=1)
one_nn.fit(X_tr, y_tr)
print(one_nn.score(X_tr, y_tr))
print(round(one_nn.score(X_te, y_te), 3))
```

Output

```
1.0
0.933
```

- On training data it just finds each flower itself
- That is memorising, not learning
- Only the test score shows real skill
- Never report training accuracy as your result

Generalisation: the real goal

1

Memorising

Learning the exact exam questions. Perfect on those, lost on new ones.

2

Generalising

Understanding the patterns, so new questions can be answered too.

3

How we check

Keep a test set the model never sees, and judge it only there.

Watch it: memorising vs generalising

WATCH FOR

- $k = 1$ wraps a border around every training point
- 100% on training data, 74% on new points
- $k = 15$ is smoother: 87% on training data
- But 88% on new points: it generalises better

MACHINE LEARNING, STEP BY STEP

Memorising vs generalising

A model that memorises its training data scores 100% on it, then stumbles on new data. Watch it happen.

Click the video to play, click again to pause. Silent: talk the class through it.

Vocabulary for the rest of the course

Word	Meaning
Feature (X)	An input column the model uses
Label / target (y)	The answer we want to predict
Model	The learned rule that maps features to a label
Training (fit)	Learning the rule from examples
Prediction	The model's answer for a new example
Hyperparameter	A setting we choose, e.g. k in k-nearest neighbours
Generalisation	Doing well on data the model has never seen

Your turn

8 minutes

- 1 Load the wine dataset: `from sklearn.datasets import load_wine`
- 2 Split it 70/30 with `stratify`
- 3 Score the same three models
- 4 Which model wins? Is it the same as for iris?

Code

```
from sklearn.datasets import load_wine
wine = load_wine(as_frame=True)
X, y = wine.data, wine.target
...
```

Starter code: replace the ... with your own code

Interactive lab: What is ML? Lab

In the lab you can

1 Sort the tasks

12 questions: supervised or not, category or number.

2 Be the model

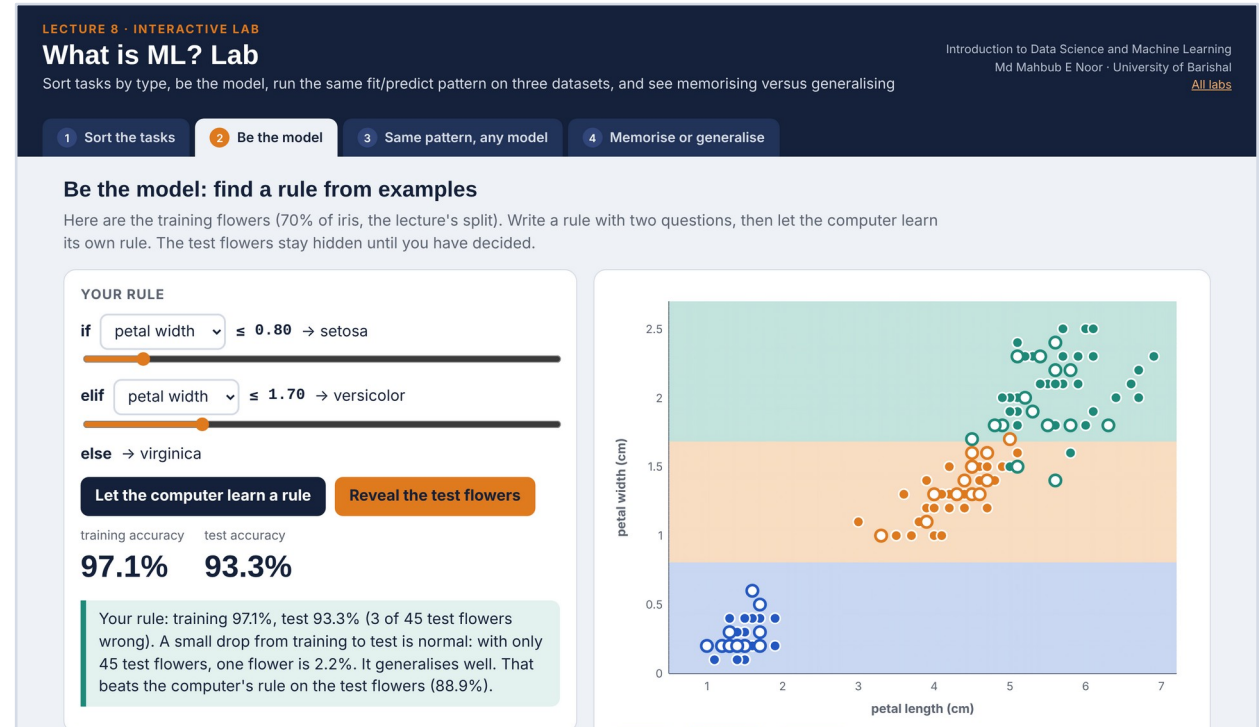
Write a two-question rule for iris, then meet the tree's.

3 Same pattern, any model

fit / score on iris, wine, breast cancer; guess the winner.

4 Memorise or generalise

k-NN from $k = 1$ to 101: training vs new points.



Open the lab ►

Works offline in any browser

Labs_open_in_browser/
L8_What_is_ML_Lab.html

Today's toolkit

supervised	Learn from labelled examples
unsupervised	Find structure without labels
classification	Predict a category
regression	Predict a number

<code>model.fit(X, y)</code>	Train
<code>model.predict(X)</code>	Answer for new rows
<code>model.score(X, y)</code>	Accuracy (classifiers)
test set	Data the model never sees in training

Homework 8

Due before Lecture 9 • [date]

1 Classify the task

Write 6 prediction questions from daily life. Label each: classification or regression, supervised or unsupervised.

2 Features and label

For two of your questions, list 3 useful features and the label.

3 Three models

Repeat the 'same pattern' slide on `load_breast_cancer`. Report the three test accuracies.

4 Train vs test

For `KNeighborsClassifier(n_neighbors=1)` on breast cancer, print training and test accuracy. Explain the gap.

5 Explain

In 3 sentences: why do we judge a model only on data it has not seen?

Submit on [LMS / Google Classroom]: your notebook file `HW8_<your ID>.ipynb`, or a Colab link set to “Anyone with the link can view”.

Recap: Lecture 8 in 90 seconds

RECAP

Lecture 8 in 90 seconds

A short recap of today's lecture. Pause any time.

Click to play or pause. Watch it again at home to revise.