

LECTURE 10

Classification

k-nearest neighbours and decision trees

Today's key line of code

```
KNeighborsClassifier(n_neighbors=3)  
DecisionTreeClassifier(max_depth=2)
```

The road ahead: machine learning

Today



Today you will

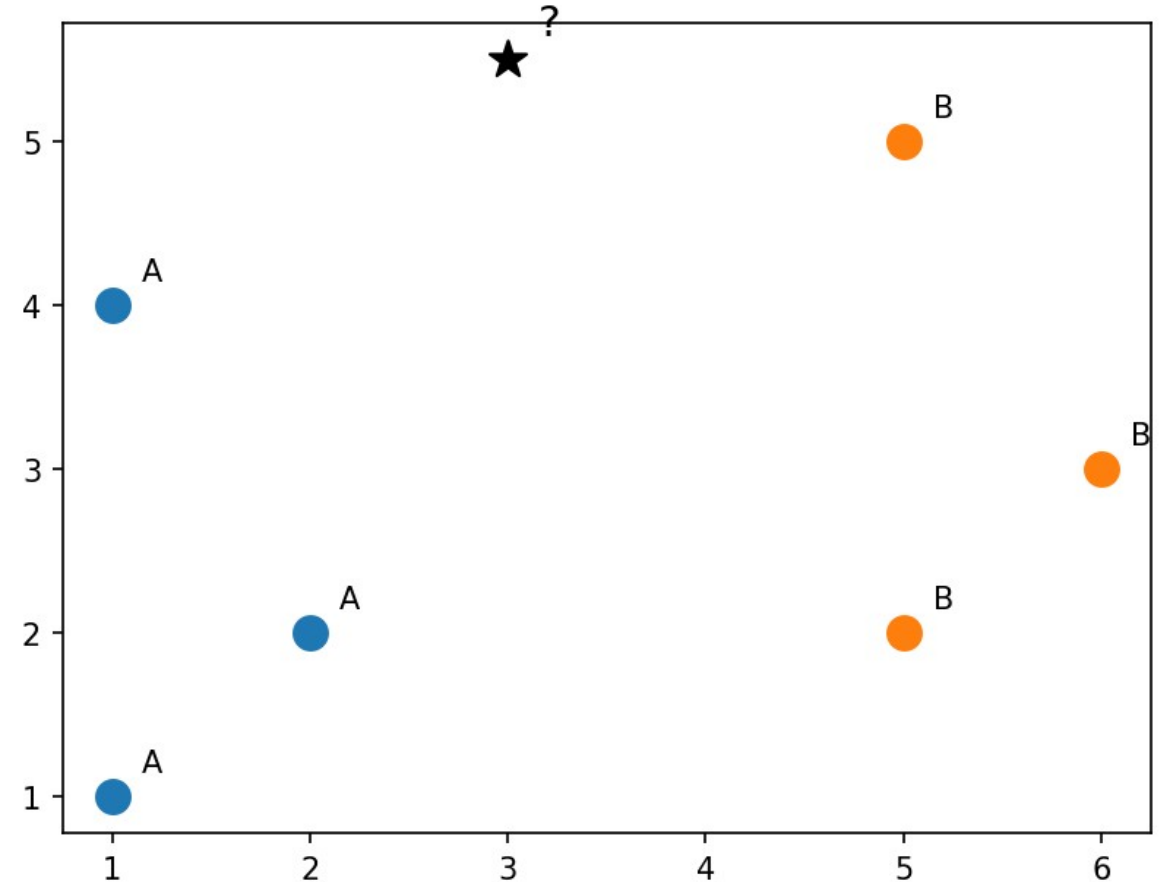
- 1 Classify a new point by hand with k-nearest neighbours
- 2 See how the choice of k changes the answer
- 3 Read the rules a decision tree learns
- 4 Follow a tree by hand to classify a flower
- 5 Compare the two models and choose sensibly

Six labelled points and one new point

Class A in blue, class B in orange. Which class should the new point (?) get?

Code

```
import matplotlib.pyplot as plt
pts = [(1, 1, "A"), (2, 2, "A"), (1, 4, "A"),
       (5, 5, "B"), (6, 3, "B"), (5, 2, "B")]
for x, y, c in pts:
    plt.scatter(x, y, s=120,
                color="tab:blue" if c == "A" else "tab:orange")
    plt.annotate(c, (x + 0.15, y + 0.15))
plt.scatter(3, 5.5, s=160, marker="*", color="black")
plt.annotate("?", (3.15, 5.65), fontsize=14)
plt.show()
```



By hand: distances to the new point (3, 5.5)

Distance = $\sqrt{(x - 3)^2 + (y - 5.5)^2}$. Sort by distance, then let the k nearest vote.

Code

```
# k = 1: nearest is B (2.06)      -> B
# k = 3: B (2.06), A (2.5), A (3.64)
#           A has 2 votes        -> A
# Same data, different k, different answer!
```

rank	point	class	distance
1	(5, 5)	B	2.06
2	(1, 4)	A	2.50
3	(2, 2)	A	3.64
4	(6, 3)	B	3.91
5	(5, 2)	B	4.03
6	(1, 1)	A	4.92

Click to reveal the next row. Predict it first!

Watch it: k-nearest neighbours voting

WATCH FOR

- A circle grows from the new point
- It stops when it holds k neighbours
- The neighbours vote; the majority wins
- With $k = 1$ and $k = 3$ the answer differs

CLASSIFICATION, STEP BY STEP

k-nearest neighbours: who is closest?

A circle grows around the new point until it contains k neighbours.
Then the neighbours vote.

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

k-nearest neighbours in scikit-learn

- `n_neighbors` is `k`
- Small `k`: follows every single point (can overfit)
- Large `k`: smoother, but may ignore real detail
- Check several `k` on the test set... for now

Code

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
X, y = load_iris(return_X_y=True)
X_tr, X_te, y_tr, y_te = train_test_split(
    X, y, test_size=0.3, random_state=1, stratify=y)
for k in [1, 3, 5, 15, 50]:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_tr, y_tr)
    print(k, round(knn.score(X_te, y_te), 3))
```

Output

```
1 0.978
3 0.978
5 0.978
15 0.978
50 0.867
```

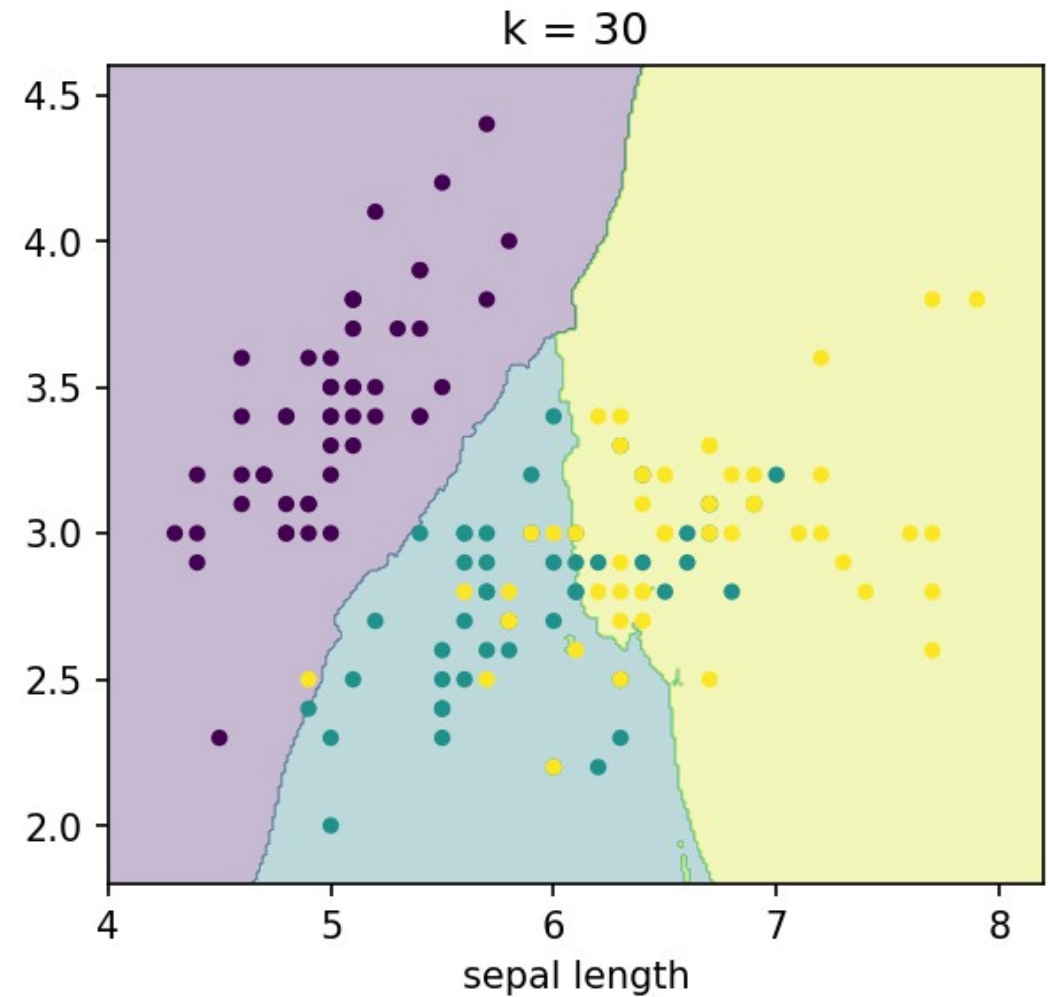
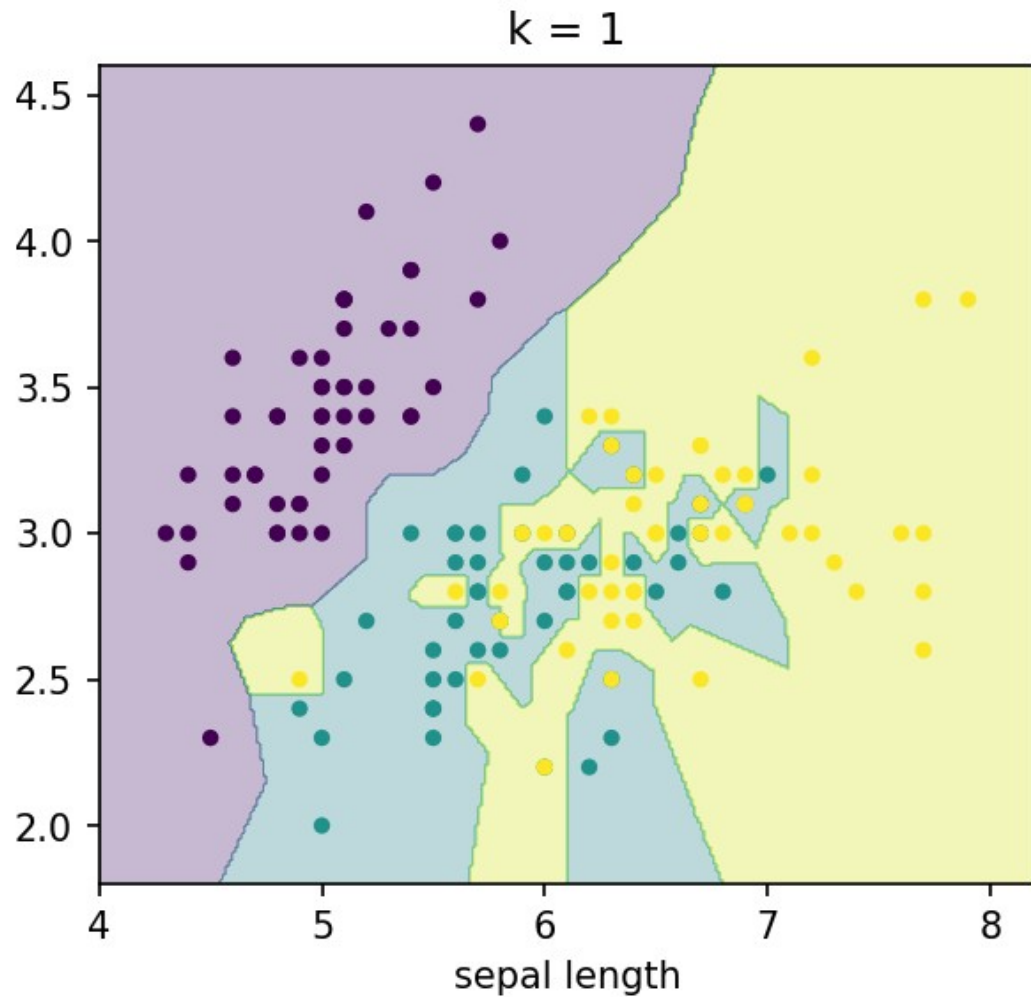
Decision boundaries: small k vs large k

Two sepal measurements, where the species overlap. Colours show the predicted class at every point: $k = 1$ draws jagged islands, $k = 30$ smooth regions.

Code

```
import numpy as np
X2 = X[:, 0:2]
xx, yy = np.meshgrid(np.linspace(4, 8.2, 300),
                     np.linspace(1.8, 4.6, 300))
grid = np.c_[xx.ravel(), yy.ravel()]
fig, axes = plt.subplots(1, 2, figsize=(10, 4))
for ax, k in zip(axes, [1, 30]):
    m = KNeighborsClassifier(n_neighbors=k).fit(X2, y)
    zz = m.predict(grid).reshape(xx.shape)
    ax.contourf(xx, yy, zz, alpha=0.3, cmap="viridis")
    ax.scatter(X2[:, 0], X2[:, 1], c=y, s=12, cmap="viridis")
    ax.set_title(f"k = {k}")
    ax.set_xlabel("sepal length")
plt.show()
```

Decision boundaries: small k vs large k: the result



A decision tree plays twenty questions

1

Ask a question

Is petal width at most 0.75 cm?
Each question splits the data in two.

2

Pick the best question

The tree chooses the question that makes each side as pure as possible.

3

Repeat

Keep asking until the groups are pure or a depth limit is reached. Leaves give the answer.

Train a tree and read its rules

`max_depth=2`: at most two questions per flower · `export_text` prints the learned rules · class 0 = setosa, 1 = versicolor, 2 = virginica · Nobody wrote these rules: the data did

Code

```
from sklearn.tree import DecisionTreeClassifier, export_text
names = ["sepal len", "sepal wid", "petal len", "petal wid"]
tree = DecisionTreeClassifier(max_depth=2, random_state=0)
tree.fit(X_tr, y_tr)
print(export_text(tree, feature_names=names))
print(round(tree.score(X_te, y_te), 3))
```

Output

```
|--- petal wid <= 0.75
|   |--- class: 0
|--- petal wid > 0.75
|   |--- petal len <= 4.75
|       |--- class: 1
|       |--- petal len > 4.75
|           |--- class: 2
```

0.956

By hand: follow the tree for three flowers

Use the two rules from the previous slide. Then check: the model's own predictions are in the last column.

Code

```
if petal_wid <= 0.75:  
    -> setosa (0)  
elif petal_len <= 4.75:  
    -> versicolor (1)  
else:  
    -> virginica (2)
```

petal len	petal wid	path	model says
1.5	0.2	wid <= 0.75	0
4.7	1.2	wid > 0.75, len <= 4.75	1
5.8	2.2	wid > 0.75, len > 4.75	2

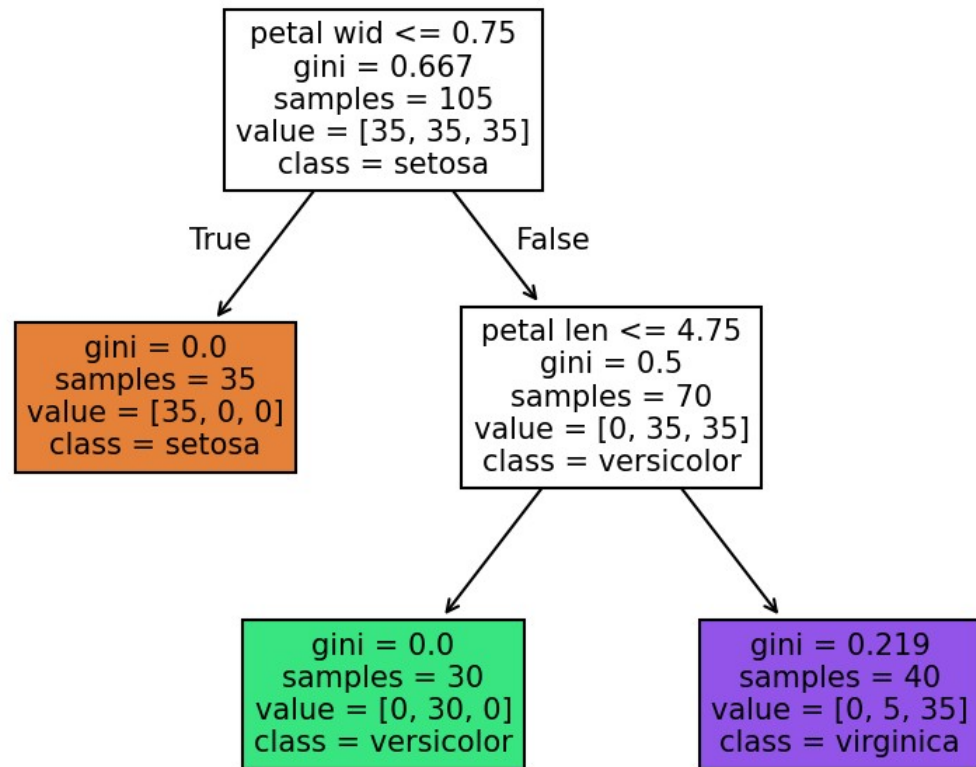
Click to reveal the next row. Predict it first!

Draw the tree

Each box shows the question, how many flowers reach it, and the class counts. Leaves at the bottom give the answer.

Code

```
from sklearn.tree import plot_tree
plt.figure(figsize=(7, 5.5))
plot_tree(tree, feature_names=names,
          class_names=["setosa", "versicolor", "virginica"],
          filled=True, fontsize=10)
plt.show()
```



Watch it: a tree splitting the data

WATCH FOR

- First question: a horizontal cut at petal width 0.75
- Setosa is separated completely
- Second question: a vertical cut at petal length 4.75
- Each region gets the majority class

CLASSIFICATION, STEP BY STEP

A decision tree splitting the data

Each question is a straight cut across the chart.
Each region gets the majority class.

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

k-nearest neighbours or decision tree?

k-nearest neighbours

- No real training: remembers all data
- Slow to predict on big data
- Needs features on similar scales
- Hard to explain a single answer

Decision tree

- Learns explicit if/else rules
- Fast to predict
- Scale of features does not matter
- Deep trees overfit easily (Lecture 11)

Your turn

10 minutes

- 1 Train DecisionTreeClassifier with max_depth 1, 2, 3, 10 on iris
- 2 Print training and test accuracy for each
- 3 Which depth would you choose, and why?
- 4 Bonus: print export_text for depth 3

Code

```
for d in [1, 2, 3, 10]:  
    t = DecisionTreeClassifier(max_depth=d, random_state=0)  
    t.fit(X_tr, y_tr)  
    print(d, t.score(X_tr, y_tr), t.score(X_te, y_te))
```

Common mistakes with classifiers

Read the last line of an error first. It names the problem.

What you typed	What Python says	What it means
<pre>KNeighborsClassifier().predict([[1, 2, 3, 4]])</pre>	<pre>NotFittedError: This KNeighborsClassifier instance is not fitted yet. Call 'fit' with appropriate arguments before using this estimator.</pre>	Not fitted yet. Call fit before predict.
<pre>knn = KNeighborsClassifier(n_neighbors=200) knn.fit(X_tr, y_tr) knn.predict(X_te)</pre>	<pre>ValueError: Expected n_neighbors <= n_samples_fit, but n_neighbors = 200, n_samples_fit = 105, n_samples = 45</pre>	k cannot be larger than the number of training rows (105).
<pre>tree.predict([[1.5, 0.2]])</pre>	<pre>ValueError: X has 2 features, but DecisionTreeClassifier is expecting 4 features as input.</pre>	The model was trained on 4 features; give it 4 values.

Interactive lab: Classifier Lab

In the lab you can

1 k-NN by hand

Click to move the new point; the k nearest neighbours vote.

2 Decision boundaries

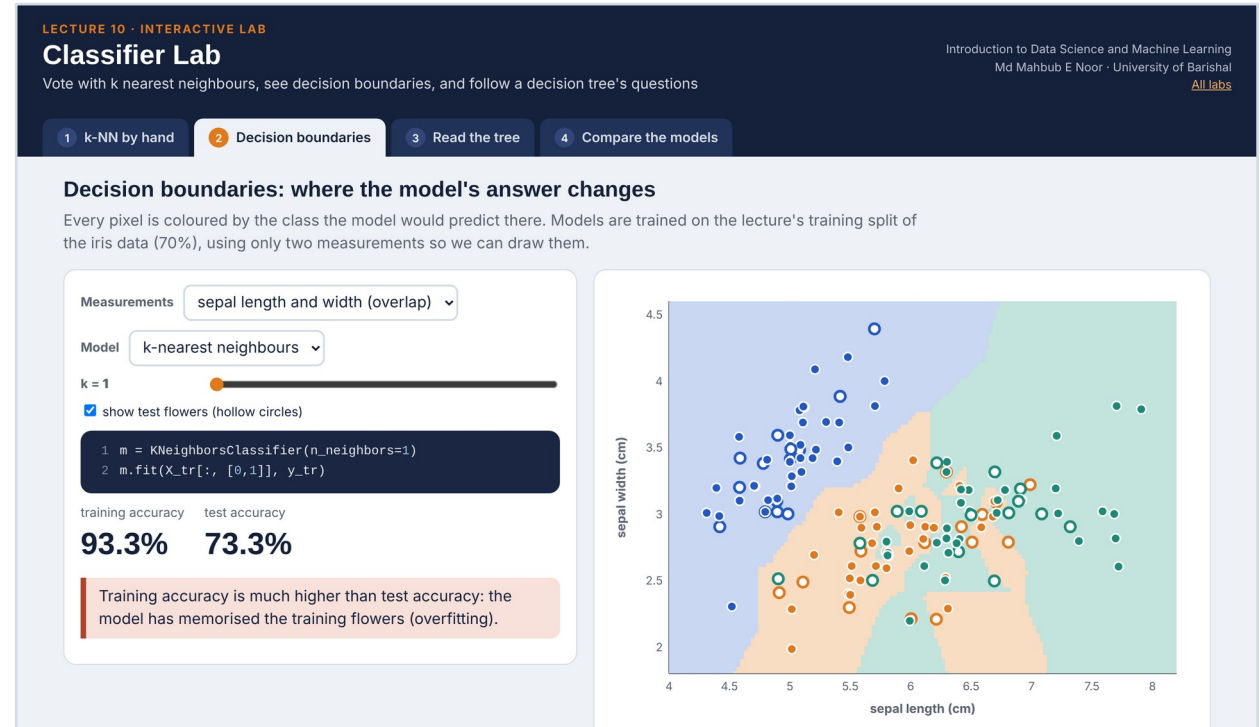
k-NN and trees on two iris measurements, train and test.

3 Read the tree

Slide a flower's measurements and follow its path.

4 Compare the models

Accuracy against k and depth, and why units matter for k-NN.



Open the lab ►

Works offline in any browser

Labs_open_in_browser/
L10_Classifier_Lab.html

Today's toolkit

KNeighborsClassifier(n_neighbors=k)	Vote of the k nearest
distance	$\sqrt{(\Delta x^2 + \Delta y^2)}$
DecisionTreeClassifier(max_depth=d)	Learned if/else rules
export_text(tree)	Print the rules

plot_tree(tree)	Draw the tree
decision boundary	Where predictions change
k, max_depth	Hyperparameters we choose

Homework 10

Due before Lecture P1 • [date]

1 By hand

For the six points and new point (4, 4), compute all distances and predict with $k = 1, 3$ and 5.

2 Choose k

On the wine dataset, print test accuracy for $k = 1, 3, 5, 9, 15$. Which k is best?

3 Tree rules

Train a depth-2 tree on wine and write its rules as if/elif statements in a text cell.

4 Boundary

Adapt the boundary plot for a decision tree of depth 2 and depth 10. Describe the difference.

5 Explain

In 3 sentences: when would you prefer a decision tree to k -nearest neighbours?

Submit on [LMS / Google Classroom]: your notebook file HW10_<your ID>.ipynb, or a Colab link set to “Anyone with the link can view”.

Recap: Lecture 10 in 90 seconds

RECAP

Lecture 10 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.