

LECTURE 13

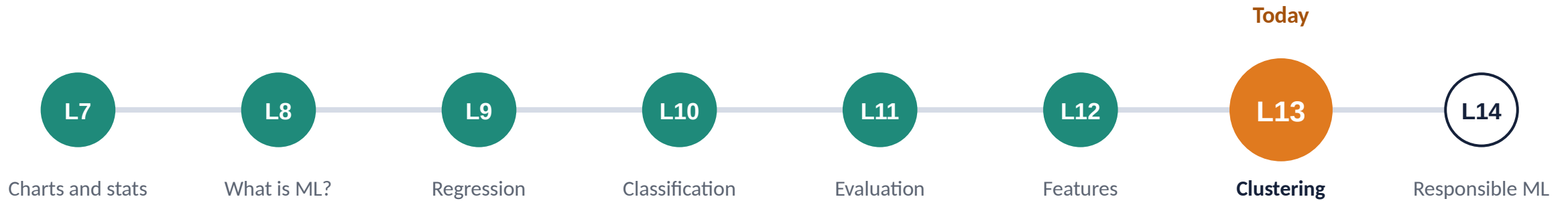
Unsupervised learning

Finding groups without labels: k-means clustering

Today's key line of code

```
KMeans(n_clusters=3, n_init=10)  
km.fit_predict(X)
```

The road ahead: machine learning



Today you will

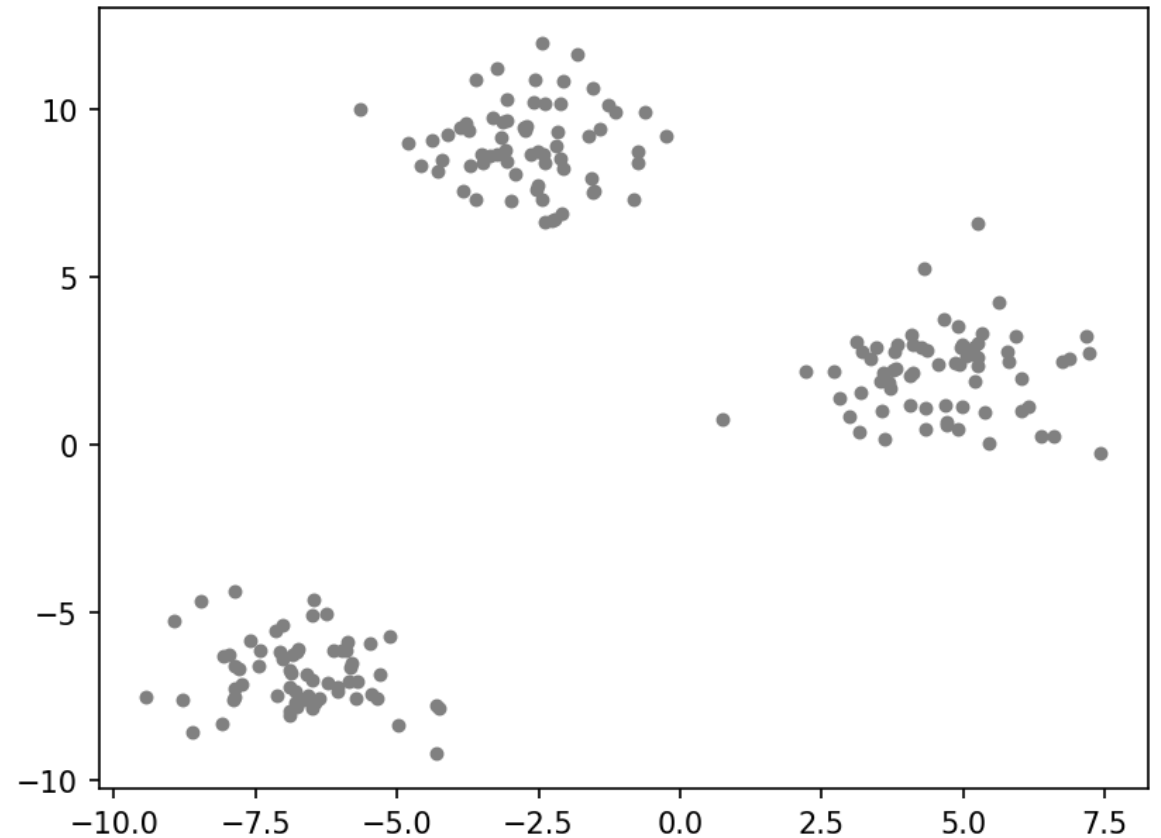
- 1 Explain what clustering is for
- 2 Run k-means by hand on a tiny example
- 3 Cluster data with scikit-learn and draw the result
- 4 Choose k with the elbow method
- 5 Know the limits of k-means

Data without labels

200 invented points. No colours, no answers. Do you see groups?

Code

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
X, _ = make_blobs(n_samples=200, centers=3,
                  cluster_std=1.2, random_state=42)
plt.scatter(X[:, 0], X[:, 1], s=14, color="gray")
plt.show()
```



k-means in four steps

1

1. Choose k

How many groups to find. We decide this; it is a hyperparameter.

2

2. Assign

Each point joins the nearest centre. Centres start at random places.

3

3. Move, repeat

Each centre moves to the mean of its points. Repeat steps 2 and 3 until nothing changes.

By hand: k-means on six numbers

Points 1, 2, 3, 10, 11, 12. $k = 2$. Start the centres at 1 and 2 (a poor start, on purpose).

Code

```
# Round 1: all but 1 are  
#   nearer to centre 2.  
#   New centres: 1 and 7.6  
# Round 2: 2 and 3 switch.  
#   New centres: 2 and 11  
# Round 3: no change: stop.
```

round	centres	group 1	group 2	new centres
1	1, 2	[1]	[2, 3, 10, 11, 12]	1, 7.6
2	1, 7.6	[1, 2, 3]	[10, 11, 12]	2, 11
3	2, 11	[1, 2, 3]	[10, 11, 12]	2, 11

Click to reveal the next row. Predict it first!

Watch it: k-means finding three groups

WATCH FOR

- Three centres start at random
- Points take the colour of the nearest centre
- Centres move to the middle of their points
- It stops when nothing changes

CLUSTERING, STEP BY STEP

k-means finding three groups

Assign each point to the nearest centre, then move each centre to the middle of its points.

Repeat until nothing changes.

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

k-means in scikit-learn

- `n_clusters` is k
- `n_init=10`: run 10 random starts, keep the best
- `fit_predict` returns a cluster number per point
- Cluster numbers are names, not an order

Code

```
import numpy as np
from sklearn.cluster import KMeans
km = KMeans(n_clusters=3, n_init=10, random_state=0)
labels = km.fit_predict(X)
print(km.cluster_centers_.round(2))
print(np.bincount(labels))
```

Output

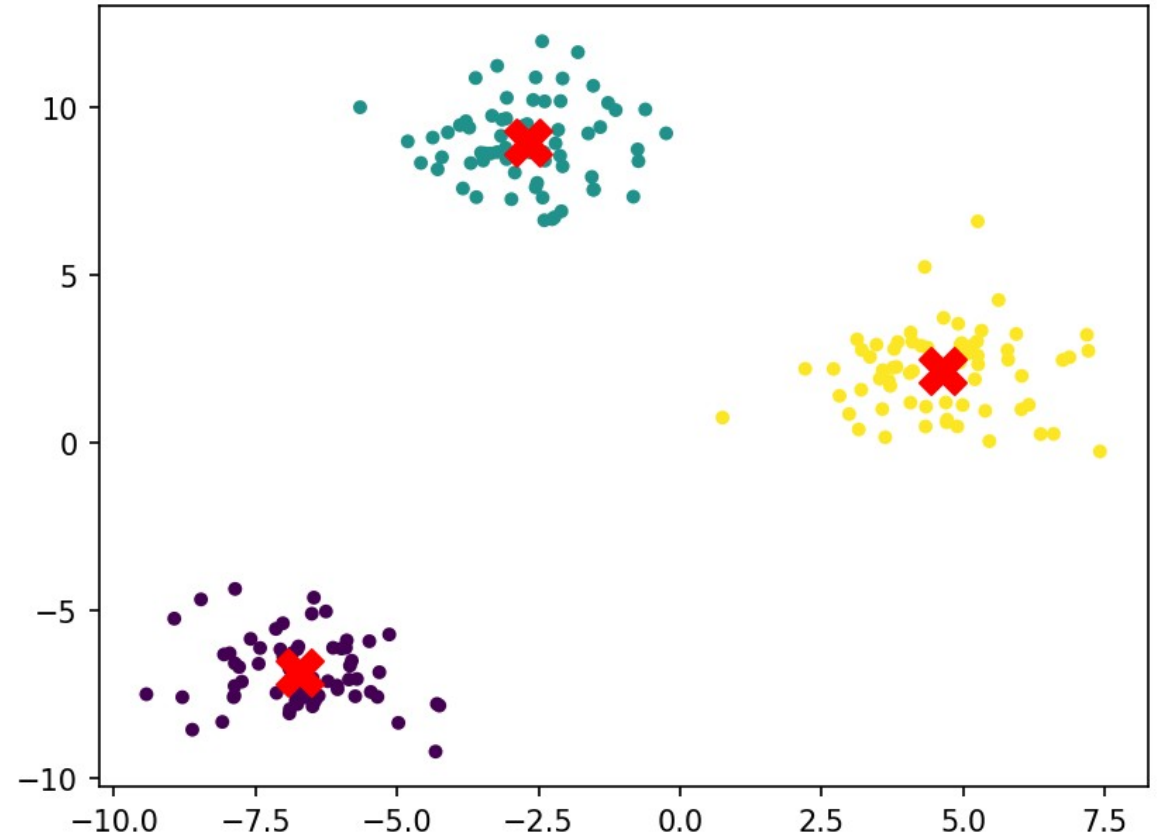
```
[[ -6.72  -6.85]
 [ -2.69   8.95]
 [  4.63   2.13]]
[66 67 67]
```

See the clusters

Colours are the clusters k-means found; crosses are the centres.

Code

```
plt.scatter(X[:, 0], X[:, 1], c=labels,  
            s=14, cmap="viridis")  
c = km.cluster_centers_  
plt.scatter(c[:, 0], c[:, 1], marker="X",  
            s=250, color="red")  
plt.show()
```

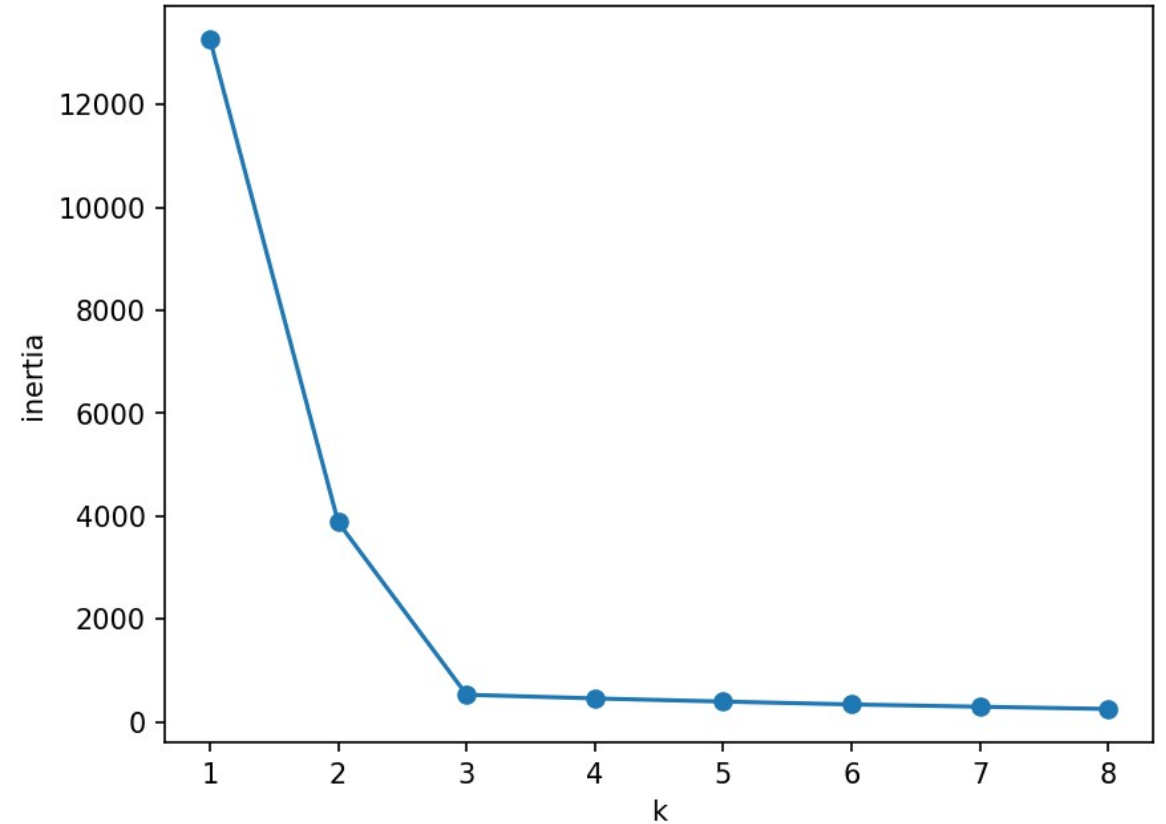


Choosing k: the elbow method

Inertia = total squared distance of points to their centre. It always falls as k grows; look for the bend.

Code

```
ks = range(1, 9)
inertia = []
for k in ks:
    km_k = KMeans(n_clusters=k, n_init=10, random_state=0)
    inertia.append(km_k.fit(X).inertia_)
plt.plot(list(ks), inertia, marker="o")
plt.xlabel("k")
plt.ylabel("inertia")
plt.show()
```



Do clusters match real classes? Iris without labels

- Cluster the 4 iris measurements, ignoring species
- Then compare clusters with the true species
- Setosa forms its own cluster perfectly
- Versicolor and virginica get partly mixed

Code

```
import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
cl = KMeans(n_clusters=3, n_init=10,
            random_state=0).fit_predict(iris.data)
species = [iris.target_names[t] for t in iris.target]
print(pd.crosstab(pd.Series(species, name="species"),
                  pd.Series(cl, name="cluster")))
```

Output

cluster	0	1	2
species			
setosa	0	50	0
versicolor	48	0	2
virginica	14	0	36

Limits of k-means

1

You must choose k

The elbow helps, but domain knowledge often decides.

2

Scale first

It uses distances, so features on large scales dominate, exactly as in k-NN.

3

Round groups only

Long, curved or very uneven groups are split badly. Other methods exist.

Your turn

10 minutes

1

Load students.csv and drop missing rows

2

Scale hours and prior_gpa with StandardScaler

3

Run k-means with $k = 3$ and add the cluster as a column

4

Print the mean hours and GPA of each cluster. Describe each group in words.

Code

```
from sklearn.preprocessing import StandardScaler
st = pd.read_csv("students.csv").dropna()
cols = ["hours", "prior_gpa"]
Z = StandardScaler().fit_transform(st[cols])
km = KMeans(n_clusters=3, n_init=10)
st["cluster"] = km.fit_predict(Z)
print(st.groupby("cluster")[cols].mean())
```

Common clustering mistakes

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

What you typed	What Python says	What it means
<pre>KMeans(n_clusters=3).fit(pd.read_csv("students.csv").dropna())</pre>	<pre>ValueError: could not convert string to float: 'Math'</pre>	Text column (dept) in the data: select numeric columns only.
<pre>KMeans(n_clusters=10).fit([[1], [2], [3]])</pre>	<pre>ValueError: n_samples=3 should be >= n_clusters=10.</pre>	Cannot make 10 clusters from 3 points.
<pre>km.predict([[1.0]])</pre>	<pre>ValueError: X has 1 features, but KMeans is expecting 2 features as input.</pre>	km was fitted on 2 features; give it 2 values.

Interactive lab: Clustering Lab

In the lab you can

1 k-means by hand

Slide 5 step by step: predict, assign, move.

2 Watch k-means

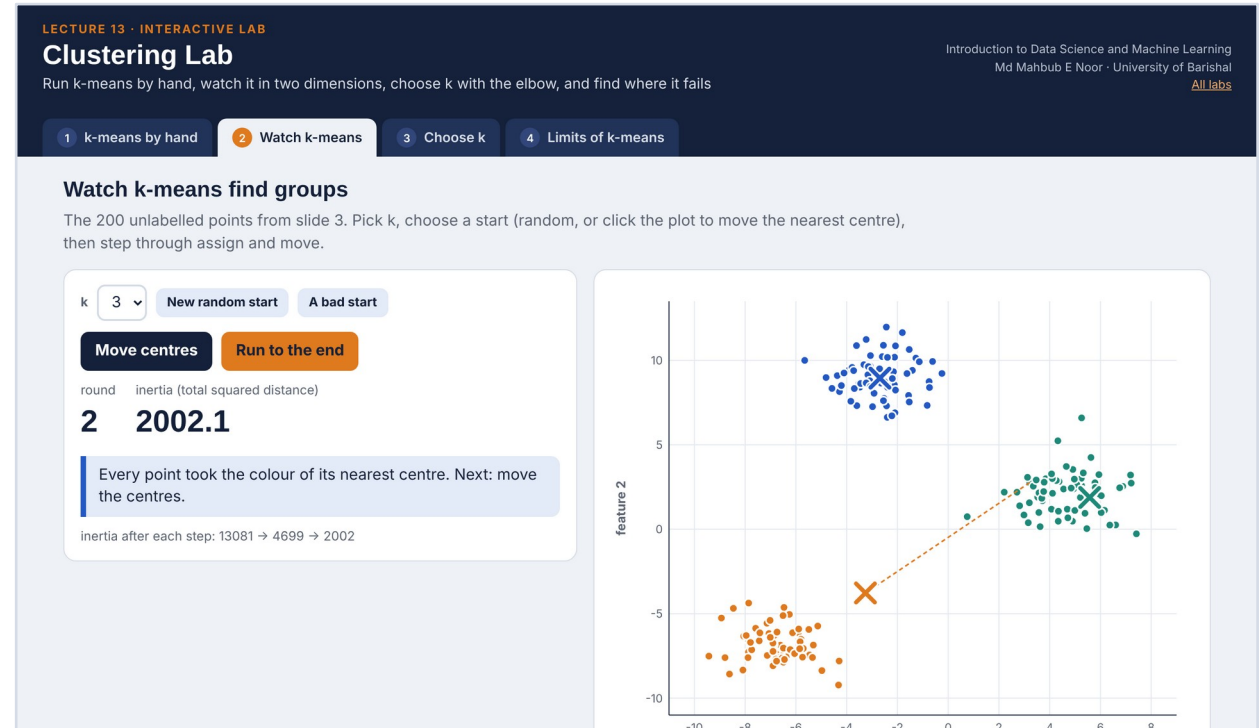
200 points in 2-D; random, clicked or bad starts.

3 Choose k

The elbow, slide 7's output, and why $n_init=10$.

4 Limits of k-means

Curved, stretched, uneven, unscaled; iris crosstab.



Open the lab ►

Works offline in any browser

Labs_open_in_browser/
L13_Clustering_Lab.html

Today's toolkit

unsupervised	No labels: find structure
KMeans(n_clusters=k, n_init=10)	k-means model
km.fit_predict(X)	Cluster number per row
km.cluster_centers_	Where the centres ended

km.inertia_	Total squared distance
elbow method	Choose k at the bend
pd.crosstab(a, b)	Compare two labelings

Homework 13

Due before Lecture P3 • [date]

1 By hand

Run k-means with $k = 2$ on 1, 2, 6, 7, 8, 20, starting with centres 1 and 20. Show each round until it stops.

2 Elbow

Make blobs with centers=4 and draw the elbow plot. Does the bend appear at 4?

3 Scaling matters

Cluster students.csv on hours and attendance with and without scaling. Compare the group means.

4 Iris

Try $k = 2$ and $k = 4$ on iris and print the crosstab. What happens to setosa?

5 Use case

Describe one real situation where clustering would help, and what you would do with the clusters.

Submit on [LMS / Google Classroom]: your notebook file HW13_<your ID>.ipynb, or a Colab link set to “Anyone with the link can view”.

Recap: Lecture 13 in 90 seconds

RECAP

Lecture 13 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.