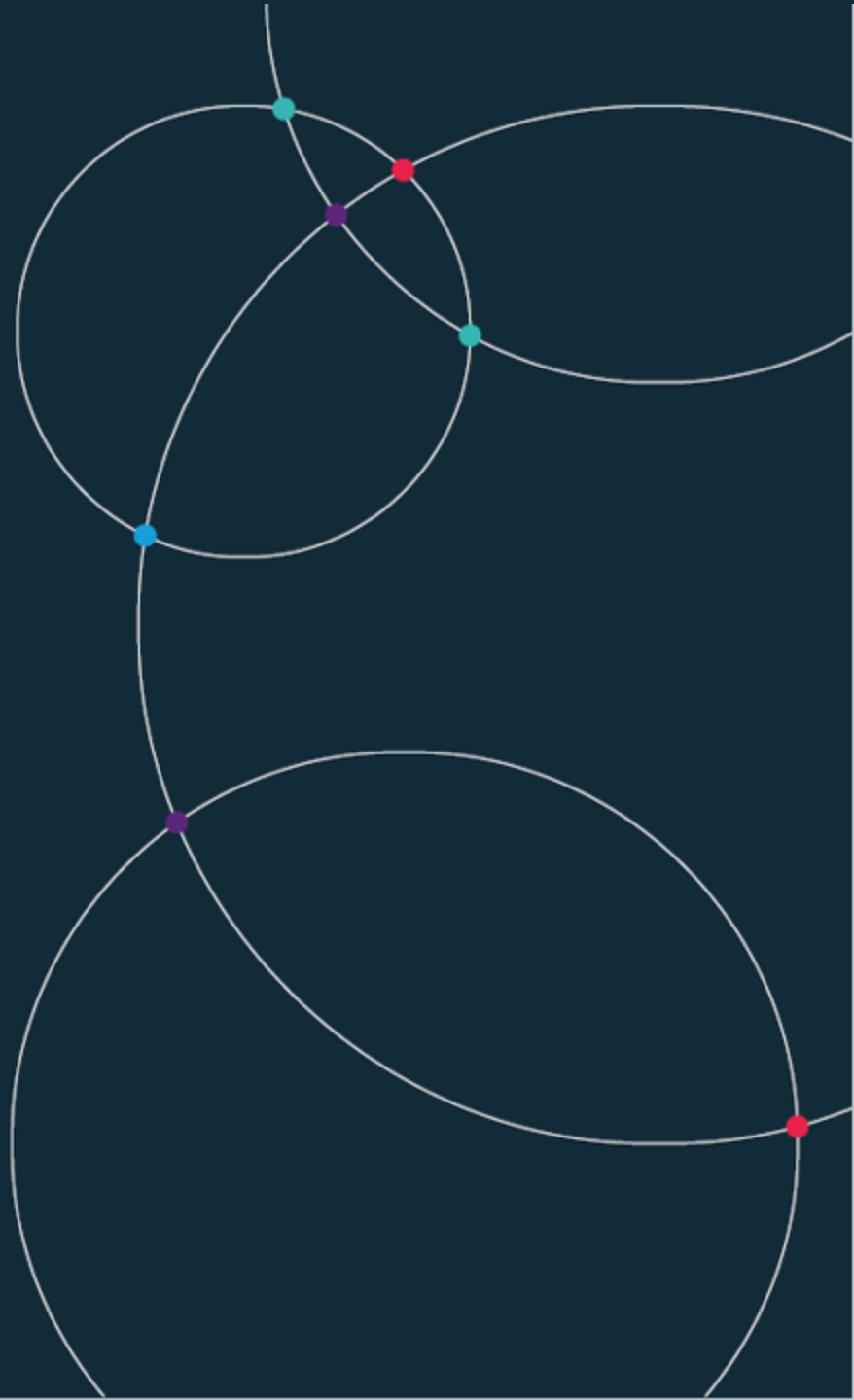
Automated Data Visua for Policymaking.

Session 10
Seminar



Resources.

Places to find, and share

Resource 1: my site:
www.richarddavies.io/data-science

Resource 2: chart library.
www.richarddavies.io/library

Resource 3: course Google sheet.
Google sheet. Link.

Resource 4: course DropBox.
DropBox. Link.

Resource 5: Playfair Prize
www.playfairprize.com

Extra session: Interactive charts

An optional recap session

Today at 15:00 - online at lse.zoom.us/j/7930269151

Recording will be added to Dropbox

Resources.

Places to find, and share

Attendance

Week 10. Reminders

Reminder: Office hours.

- Richard: Thursday, 14:10-15:00 (CBG 5.02 - book on Student Hub)
- Josh: Monday, 14:00-15:00 (CBG 5.30 – drop in)
- Finn: Tuesday, 10:30-11:30 (CBG 5.30 – drop in)
- Hannah: Wednesday, 15:00-16:00 (CBG 5.30 – drop in)

Reminder: Portfolio tasks

These are set each week and make up 20% of your grade. They can be found in the course DropBox. The file is **here**

Project & marking.

Recap

Assessment. {Deadline: 7th January}

Your project (80%)

This page sets out the student's data science project. There are weekly on-line sessions in which students can discuss ideas with the teaching team. The project consists of between 5 and 8 charts, tables or visualisations. Students briefly discuss four topics: the aims, the data, analytical challenges, conclusions. Key marking criteria include: accessibility, empirical design, data approach, automation, interactivity, clarity of writing.

More detail
Your project page should be hosted on a file named "project.html". You pick this topic yourself, and can discuss ideas with the teaching team. The project will consist of between 5 and 8 charts or visualisations. The written text that accompanies your project should not exceed 800 words in total.

You should briefly discuss four topics:

1. **Aims**: the goal of your project;
2. **Data:** the sources you used, how you accessed it, including notes on automation and/or replication;
3. **Tools:** you approach to data cleaning and/or analysis, the tools you used, any challenges you faced, and how you overcame them;
4. **Conclusions**: what you take form all of this.

Marking criteria
Your project will be marked against the follow criteria:

1. *Open-data approach.*
   - Public accessibility: hosting on a public GitHub page, with clear guidance to your repository.
   - Sourcing: links to data and notes in you code.
   - Technical guidance: description of any technical matters (APIs, scrapers) that would allow others to build on your work.

2. *Data design.*
   - Clarity of question.
   - Link between data question and datasets sought out.
   - Empirical approach: data gathering, cleaning, and analysis.

3. *Automation and replication*
   - How easy to update is your project, for you in the future, or for someone else looking to build on it.
   - The use of APIs, where suitable
   - The use of scrapers, if appropriate, and clear linking to codebooks.

4. *Visual impact and Grammar of Graphics*
   - For simple charts, your adherence to the principles of good visualisation.
   - For complex charts, their suitability and impact on the reader.
   - The use of interactives to engage the reader

5. *Description / write up*
   - Clarity of discussion
   - Logical steps from question to data, to method, to conclusion
   - Impact: does the project stimulate further work?

Grades also take into account the overall level of ambition of students' projects.

Useful web sites

The skills you will learn on the course are used by the Economics Observatory and LSE Growth Lab teams in their daily work. Prospective students are encouraged to look at these sites for a taster of the skills you will learn. Students are also given the opportunity to enter their work in the Playfair Prize, which has its own dedicated site.

www.economicsobservatory.com

www.lse.ac.uk/growth

www.playfairprize.com

Updated course guide on Moodle

Classification & Clustering.

KNN & K-Means

K-Nearest Neighbours.

Simple classification model with KNN

How it works: Classifies new points based on the majority vote of their k nearest neighbours

- Find the k closest points in your training data
- Assign the most common class among those neighbours
- "Tell me who your neighbours are, and I'll tell you who you are”

Example: Predicting US region from socioeconomic indicators

- Features: Median income ($) and firearm death rate (per 100,000)
- k=3: Look at 3 nearest states to classify Northeast vs South

→ Distance matters: Uses Euclidean distance $\sqrt{[(x_1-x_2)^2 + (y_1-y_2)^2]}$

KNeighborsClassifier Scikit-learn docs here

KNN: Scaling problem.

Introduction to scaling / normalising in ML

In KNN, classify on nearest points by Euclidean distance
$\sqrt{[(x_1-x_2)^2 + (y_1-y_2)^2]}$

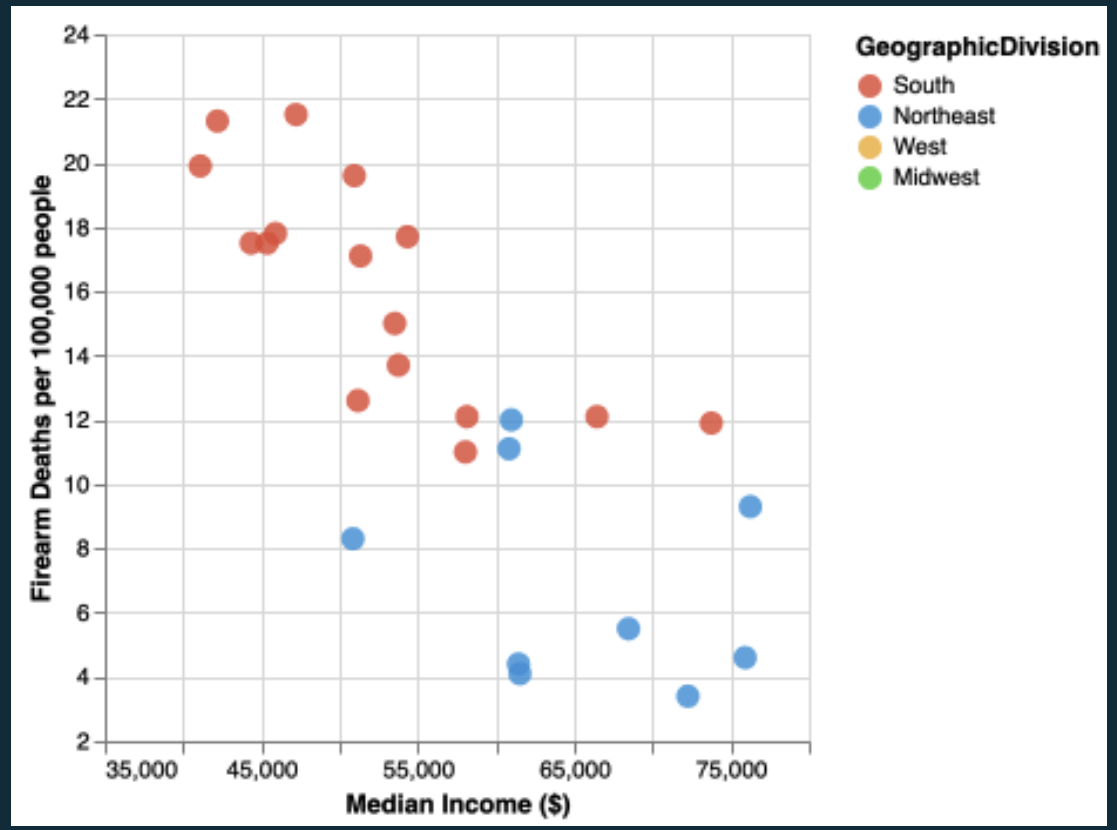
Problem: radically different scales break KNN

Example:

- Income: $40,000 - $90,000 (range ~50,000)
- Death rate: 3 - 25 per 100,000 (range ~20)
- Income differences dominate: $5{,}000^2 = 25{,}000{,}000$ vs $5^2 = 25$

Without scaling, our decision boundary is essentially a vertical line - only income matters.

Solution: Feature Scaling



Feature scaling.

Standardisation is a common requirement for ML methods

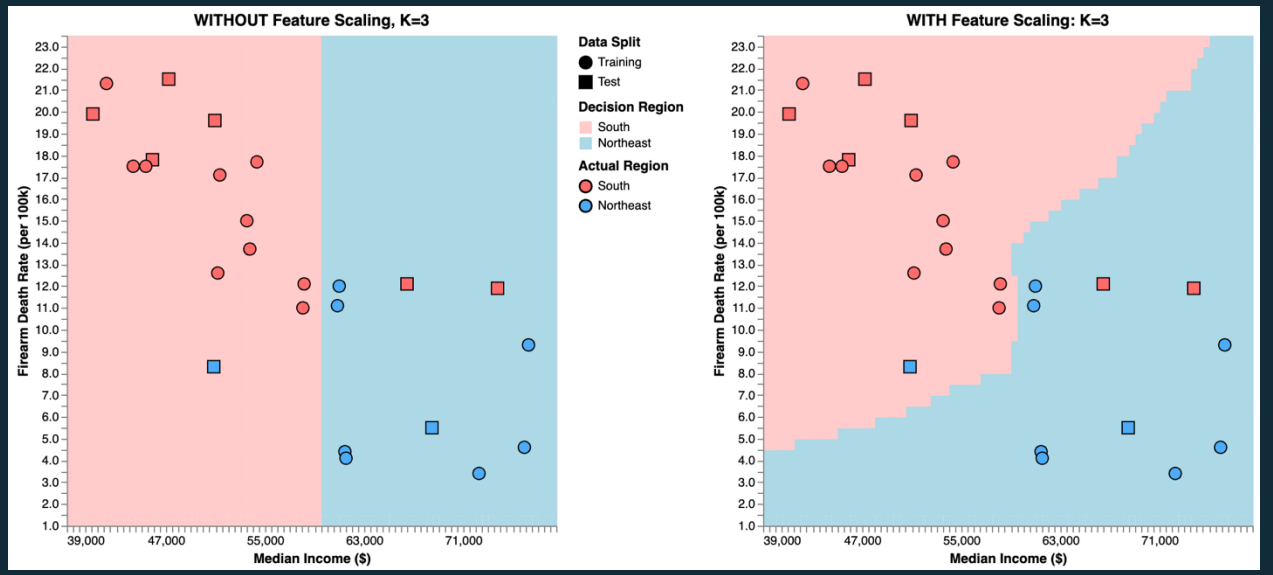
StandardScaler: transform all features (variables) to mean=0, standard deviation=1.

Now each feature contributes equally to distance.

Alternative scaling methods:

- MinMaxScaler: Scale features to [0,1] range
- RobustScaler: Uses median/IQR, robust to outliers
- Normaliser: Scale each sample to unit length

Always scale for distance-based algorithms (KNN, K-means, SVM), neural networks, gradient descent.



KNN US states example (k=3): before and after scaling firearms and income data.

See a comparison of how different scaling methods affect data distribution here.

Scikit-learn docs: StandardScaler

Unsupervised learning.

Intuition and examples

In unsupervised learning, the data is not labelled
The algorithm finds patterns in the data without your help

Examples include:

- Clustering
- Dimensionality reduction (Principal Component Analysis)

Correct answers are not needed

The data is unlabelled. It is cheaper, and can be larger.

Unsupervised learning.

When to use?

Similarity and distance between data points play a predominant role.

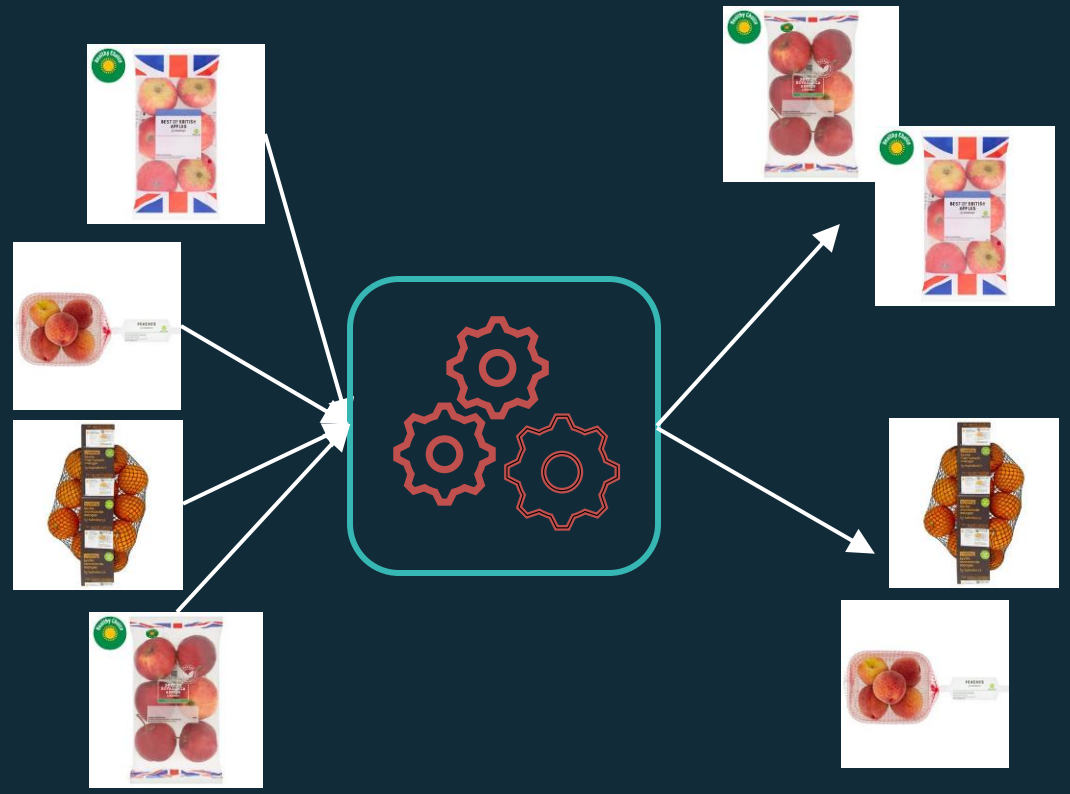
Information retrieval:

- Cluster documents & retrieve similar documents
- Features: set of words in vocabulary

Fraud detection:

- Cluster transactions and flag outliers
- Features: location, time of day etc

Ideas?



Identify unknown patterns in the feature space *V* based on unlabelled data.

K-means.

Searching for similar groups

Colab

DS10_6_Unsupervised1_KMeans1.ipynb

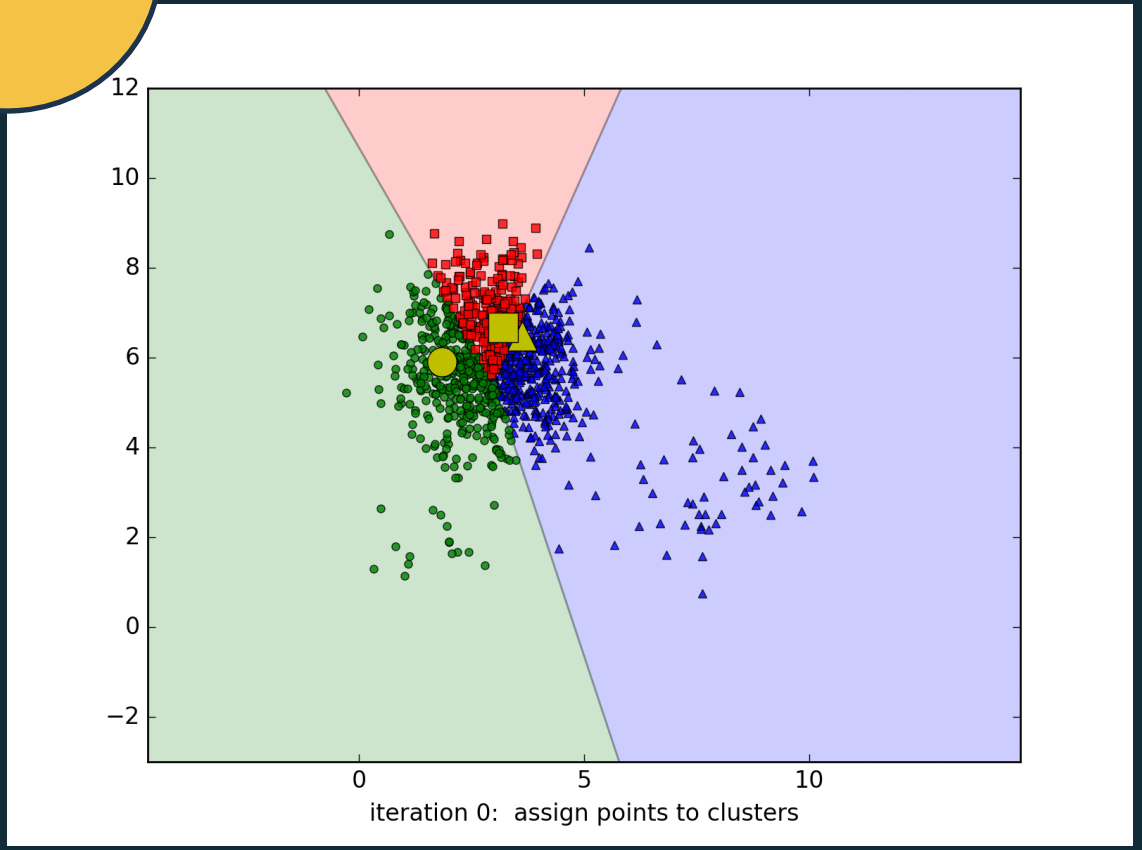
- Partitions data into ‘k’ clusters based on feature similarity
- Iterative process:
  1. Assign each point to nearest cluster centre (centroid)
  2. Recalculate centroids as mean of assigned points
  3. Repeat until convergence

- Requires choosing k upfront.

- The algorithm iteratively assigns points to nearest cluster centre, then recalculates centres.

- Minimises within-cluster variance using distance metrics.

- The initial point selection can affect final clusters.



Machinelearningcoban.com