# CAB420: Clustering and K-Means

WHAT, WHY AND (ONE APPROACH FOR) HOW

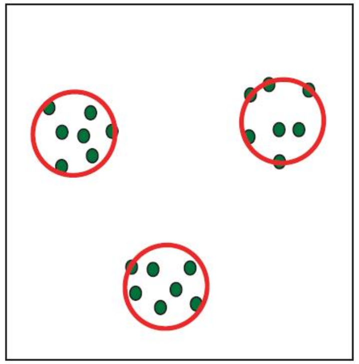
# What is Clustering?

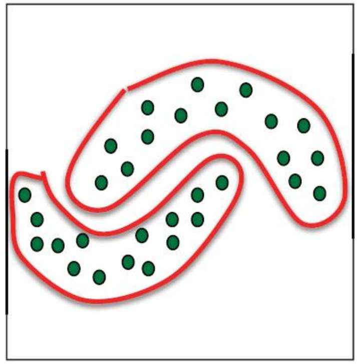
An unsupervised learning method

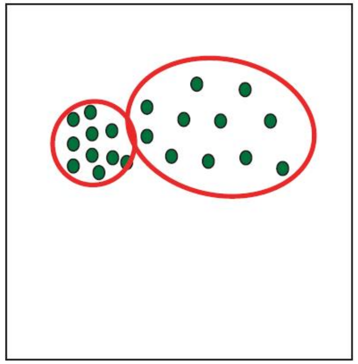
- Discover patterns in the data
- Group related points into groups

But

- What makes points related?







# Clustering Data

Why cluster data?

- To simplify it
  - Go from thousands or millions of points to a small set of cluster centres
- To find patterns or relationships
  - Find points that are similar
- To find things that are abnormal
  - i.e. points that don't fit with the rest of the data

There are lots of clustering methods

- We'll look at K-means and GMMs first

# K-Means

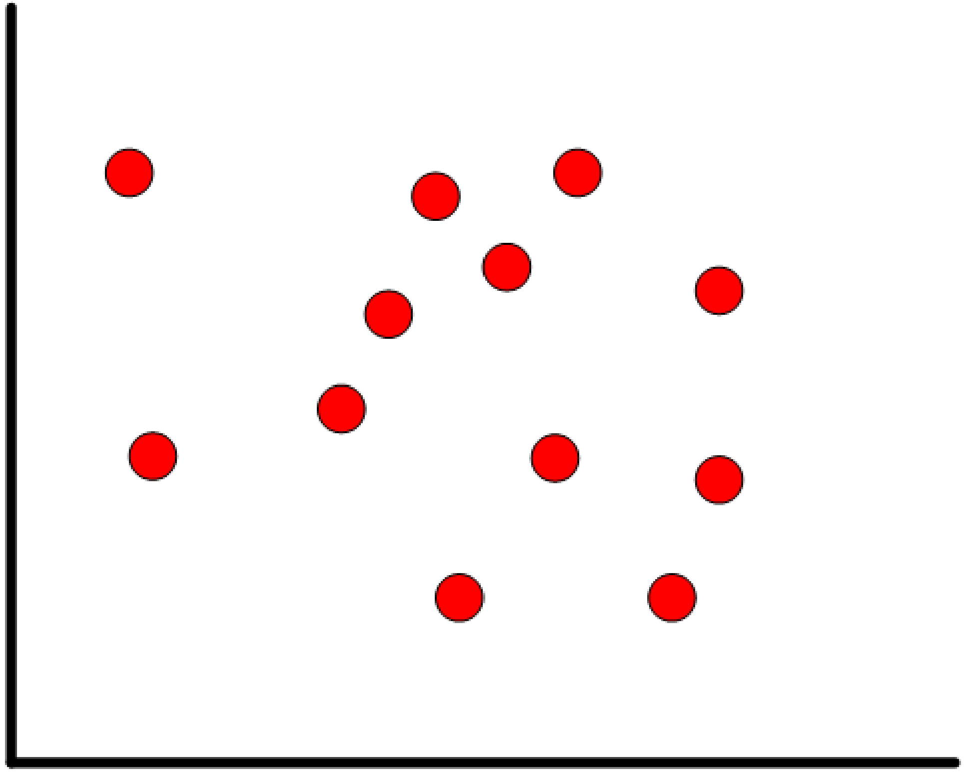
EXPLAINED

# K-Means Clustering

- Starts with a dataset and a target number of clusters, K
- Aims to find a set of K clusters that minimises the distance between each point and its cluster centre

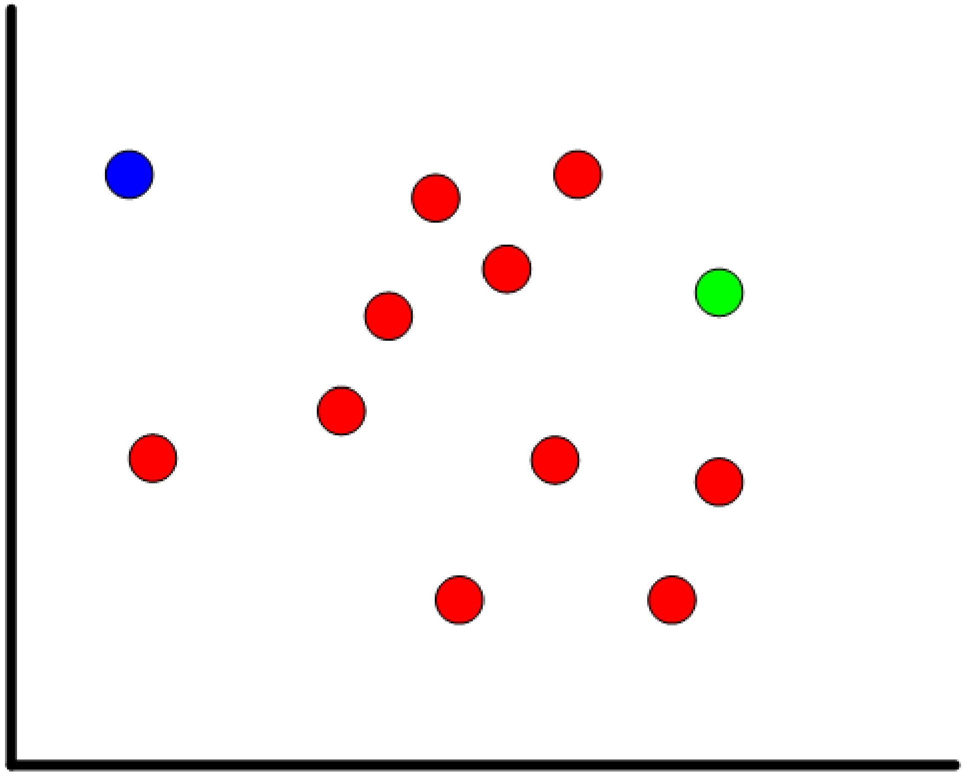
# K-Means Clustering

- Start with a set of points, and a target number of clusters, K
  - K = 2



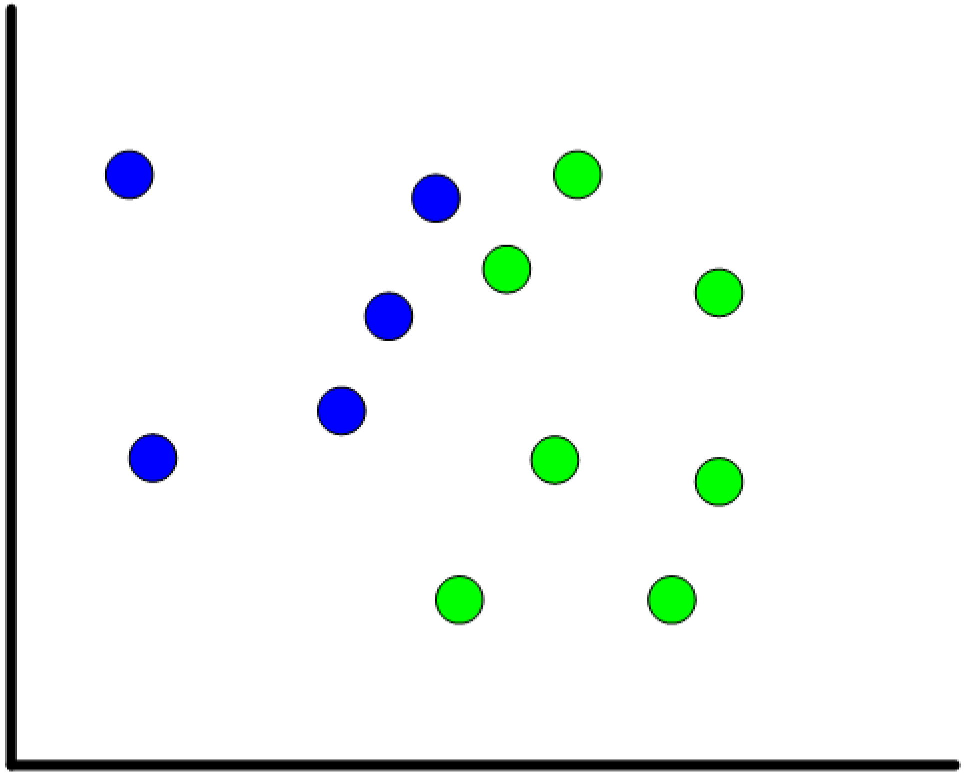
# K-Means Clustering

- Pick two points at random to be our initial cluster centres



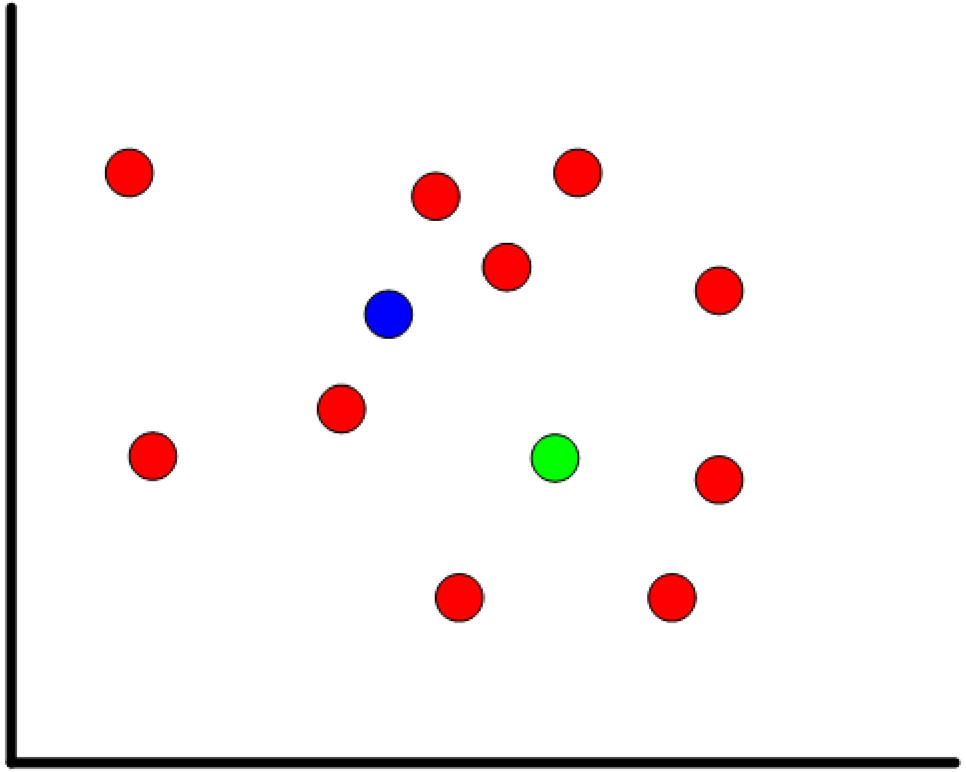
# K-Means Clustering

- Assign every point to its nearest cluster centre



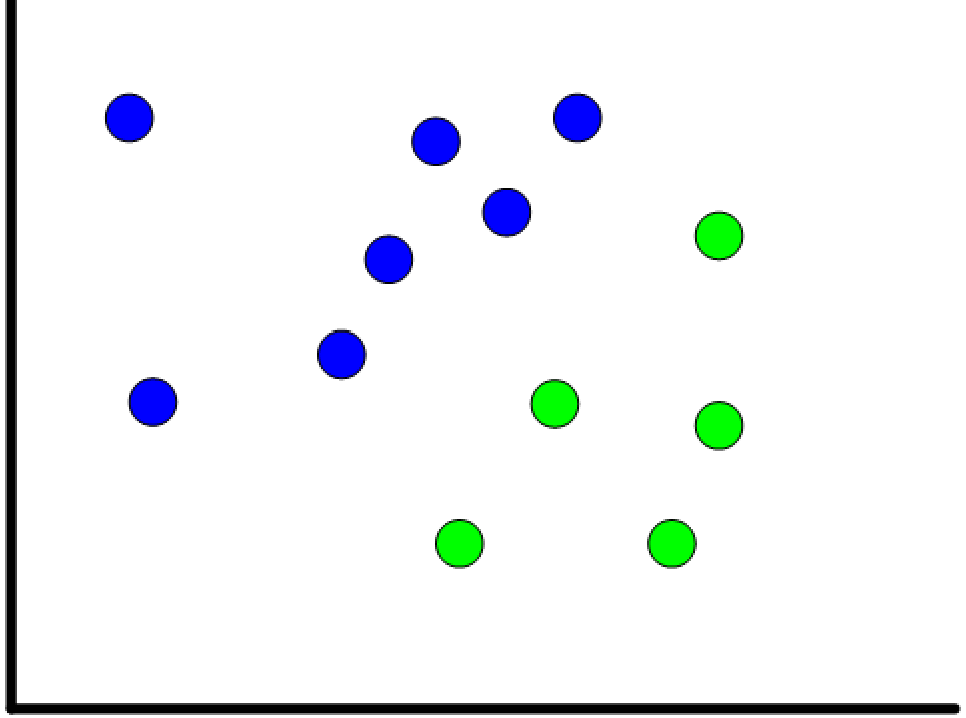
# K-Means Clustering

- Calculate new centre points based on the initial clustering, and run again



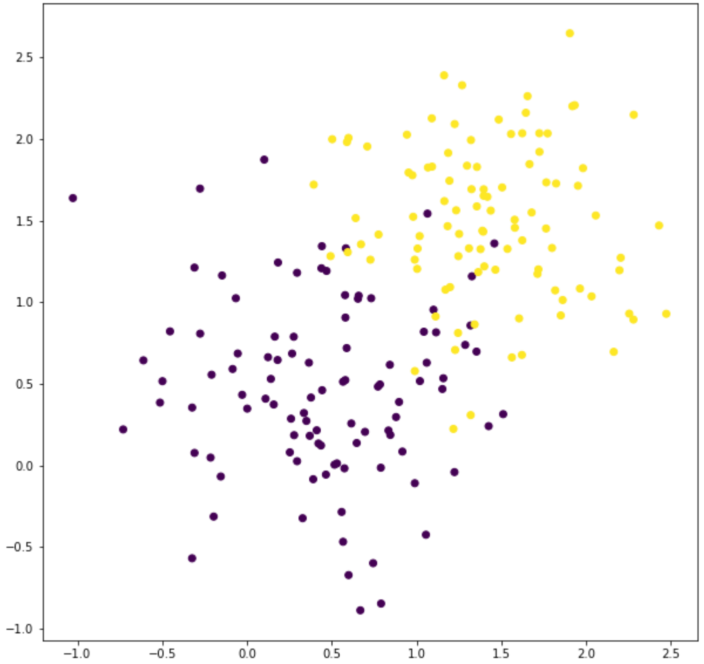
# K-Means Clustering

- We keep doing this until
- The result has converged, i.e. it’s stopped changing, or is only changing by a very small amount
- We reach a total number of iterations



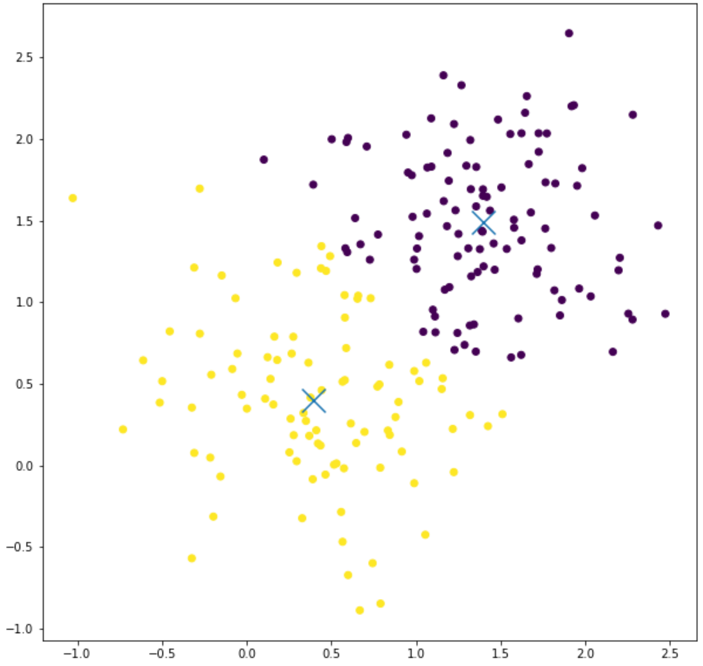
# An Example

- See ***CAB420_Clustering_Example_1_KMeans_Clustering.ipynb***
- We'll cluster some random data
  - Data contains two actual clusters
    - True cluster centres are
      - (0.5, 0.5), the first 100 points are in this cluster
      - (1.5, 1.5), the second 100 points are in this cluster
  - Clusters have some overlap



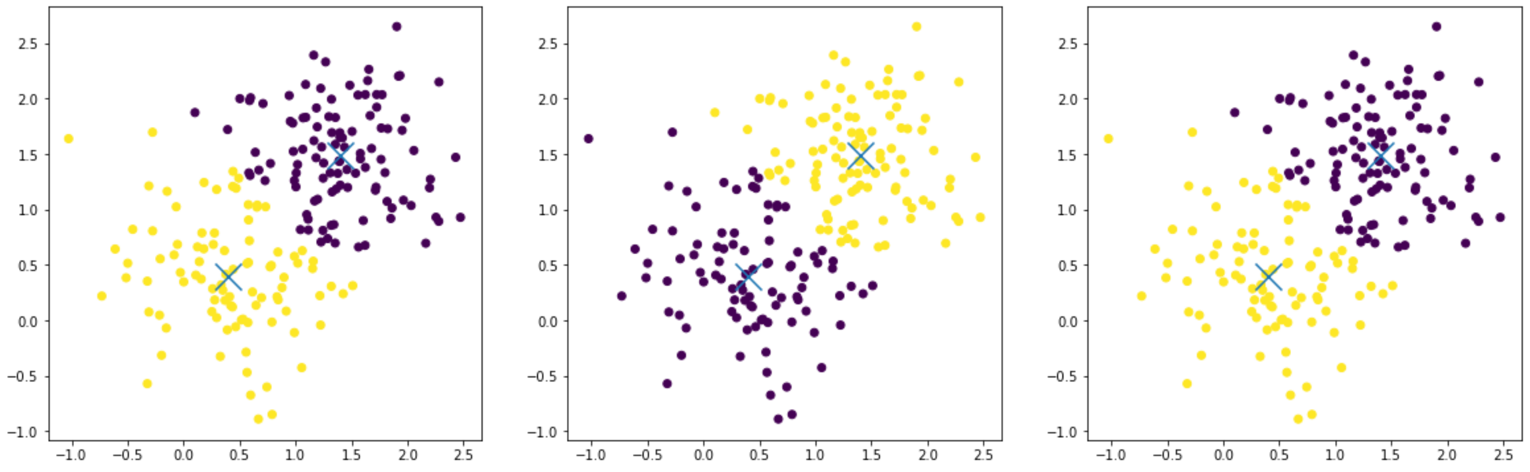
# Clustering Results

- Cluster centres estimated as:
  - (1.4003926, 1.49235897)
  - (0.39658001, 0.39824299)
- Cluster assignment is fairly accurate
  - Estimated centres are close to true centres
  - Some points at the boundary are grouped into the "other" cluster
    - This is not necessarily an error or problem, and is expected in this case



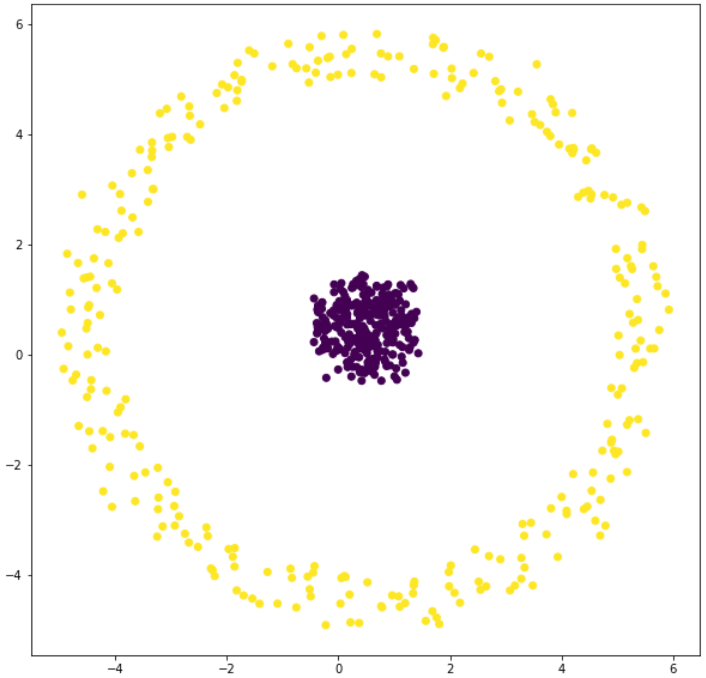
# K-Means and Randomness

- Recall, K-Means starts from random initial cluster centres
  - Different starting points lead to different results
  - Differences are small in this case due to fairly simple data
    - Differences will be more pronounced for larger values of K and more complex data



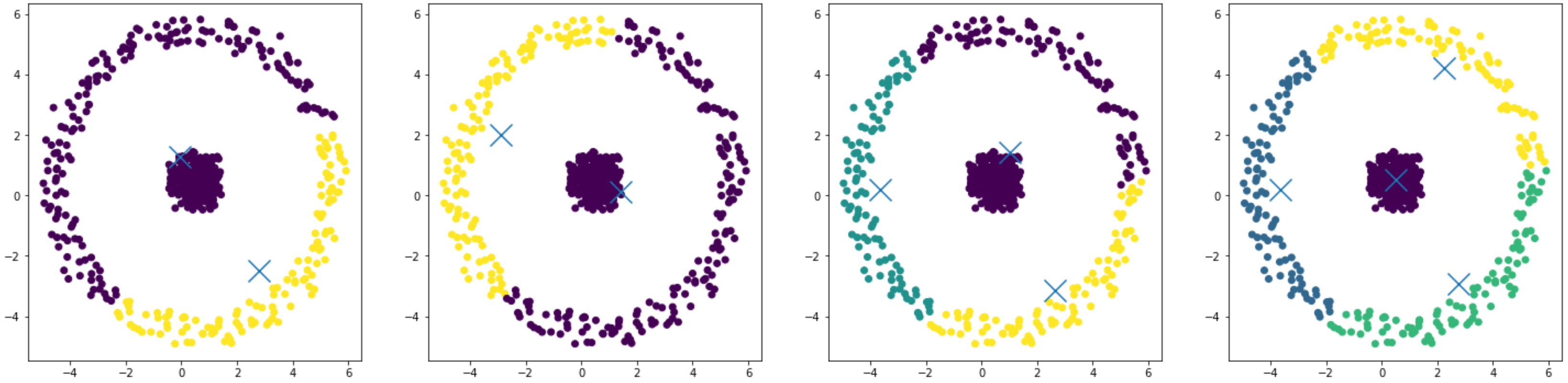
# A Second Test Case

- Two true clusters again
  - Both with a true centre of (0, 0)
  - Clear physical separation between clusters



# Clustering Results

- For K=2, we cannot recover the true clusters at all
- For K=4, we can recover the true centre cluster, but the outer ring is broken into three clusters
  - Over-clustering



# Things to Consider

WITH K-MEANS IN PARTICULAR, NOT IN GENERAL

# K-Means and Randomness

The initial cluster centres are random

- Or semi-random in the case of K-Means++

This means:

- Different runs may give us different results
- Differences will become more pronounced with
  - Bigger, more complex datasets
  - Fewer iterations
  - More clusters

# K-Means and Cluster Shapes

- K-Means extracts spherical clusters
  - Or circular in a 2D case
- Clusters will typically all have similar shapes and sizes
- Clusters cannot overlap
  - K-Means uses hard assignment, a point either belongs to a cluster, or does not

# Distance Metrics

- We can cluster any type of data
  - Just need to define a way to calculate the distance between points
- Common Metrics
  - Euclidean distance (L2)
  - Manhattan distance (L1)
  - Cosine distance (angle between points)
  - Hamming Distance (used for binary data)
- Distance metrics can have a big influence on data
  - Use the right metric for your data
- Python (sklearn) is very limited in what distance metric you can select
  - pyclustering may be worth considering if you need to change metrics

# k-Means Distance Scaling and Standardisation

- The distance metrics have an important note attached to them; comparable distance.
- If a dimension of our data has a large scale, the results of our clustering may be completely dominated by that scale.
- In cases where this is apparent, the data must be scaled in certain situations to ensure that no one variable controls the clustering.
- This is done through normalisation of data:
  - Scaling: Changing the minimum data point to 0 and maximum to 1, or
  - Standardisation: Changing the mean to zero and standard deviation to 1.

# k-Means Drawbacks

- Remember that no clustering method is the best by default.
- k-means clustering is restricted to roughly **spherical** clusters.
- Consider a ’ring’ of data around a small cluster of data. Another method can deal with clusters like this.
- k-means clustering restricts objects to **one specific cluster**.
- If a point is on the border between two clusters, there is a chance it could belong to either. Another method can deal with cases like this.

# Gaussian Mixture Models (GMMs)

ALSO EXPLAINED

# K-Means and ‘Hard Decisions’

With K-Means, each point is assigned to one cluster

- What if the point is at the boundary of two clusters?
- Wouldn’t it be better to say the point if 51% cluster 1 and 49% cluster 2?

# Gaussian Mixture Models

- A Gaussian mixture model assumes that each cluster has its **own** normal (or Gaussian) distribution with parameters $\mu_c$ and $\sigma_c$.
- Each cluster has a **weight**, $\pi_c$, included to prioritise certain clusters
  - Clusters with a higher weight have more points in them, or are more likely
- We can instead calculate the probability that a point belongs to a certain cluster.
- This probability is based on several parameters
  - The mean of each cluster, $\mu_c$
  - The covariance between variables, $\Sigma_c$
  - The weight term, $\pi_c$
- We can find the optimal parameters for this model using **maximum log likelihood estimation**.

# Log Likelihood Function

- From your regression knowledge, it is clear that different values of **parameters** lead to different data **predictions**.
- In statistics, and in our case machine learning, we would like to know the distribution that most likely gave the data points we are observing.
- These distributions rely on parameters, such as mean and variance, much like regression models rely on slopes and intercepts.
- Maximum likelihood estimation is a method that will allow us to calculate distribution parameters that dictate the shape of the distribution.

# Log Likelihood Functions

- If each data point is generated independently of the others, then the total probability of observing our data is the product of the probability of observing each single data point separately.
- Thus, the likelihood function is given by:

$$P(x|\theta) = \prod_{i=1}^{n} \Pr(x_i)$$

where $\theta$ is the set of parameters in a distribution and $\Pr(x_i)$ is the **probability mass function** of the distribution which gives our data points.

- These expressions can be difficult to differentiate (which is necessary to optimise), so we take the **log** of the function to simplify the product:

$$L\big(P(x|\theta)\big) = \sum_{i=1}^{n} \log(\Pr(x_i))$$

# Log Likelihood Functions

- Once we simplify the log likelihood function, we **differentiate** it with respect to each of the parameters.
- In order to find the **maximum** likelihood estimate for each parameter, we set the derivative to 0 and solve.
- In machine learning, typically optimisation requires finding minimum values (i.e., minimising errors).
- We use **negative log likelihoods** (NLLs);

$$NL\big(P(x|\theta)\big) = -L(P(x|\theta))$$

as a way to ensure that this process will always result in a minimum.

- If we set out to **minimise** the likelihood estimator, we will now find the **actual maximum**, rather than some local minimum likelihood estimate which would be the opposite of our goal!

# Gaussian Mixture Models

- The probability mass function for a data point in a Gaussian mixture model is given by:

$$\Pr(x) = \sum_{c=1}^{K} \pi_c N(x|\mu_c, \Sigma_c)$$

where $\sum_{c=1}^{K} \pi_c = 1, 0 \leq \pi_c \leq 1$, and $x$ has multiple dimensions based on the terms in the clustering

- The negative log likelihood that we must minimise for optimal parameters is

$$-\log(\Pr(x|\pi, \mu, \Sigma)) = -\sum_{i=1}^{N} \log(\sum_{c=1}^{K} \pi_c N(x|\mu_c, \Sigma_c))$$

- Difficult to solve due to the sum inside the logarithm.
- Optimisation is instead achieved using an iterative technique: the expectation maximisation (EM) algorithm

# Learning a GMM

Expectation-Maximisation (EM) Algorithm

- Iterative, two step process
- Expectation Step
  - Determine likelihoods for each sample for current model
- Maximisation Step
  - Update model parameters

# Learning a GMM – Starting Conditions

We need an initial set of clusters

- Use K-means to create an initial clustering result
  - Mean $\mu_c$
  - Covariance $\Sigma_c$
  - Weight (or size) $\pi_c$

Good initialisation is important

- EM will converge to a maximum
- Need to a good initialisation to avoid getting stuck in a local maximum

# Learning a GMM - Expectation

For each data point

- Compute $r_{ic}$, the probability that it belongs to cluster $c$
- Normalise to sum to 1

$$r_{ic} = \frac{\pi_c N(x_i; \mu_c, \Sigma_c)}{\Sigma_{c'} \pi_{c'} N(x_i; \mu_{c'}, \Sigma_{c'})}$$

- Points with a good fit to a mode will have a high weight

# Learning a GMM - Maximisation

Start from the assignment probabilities, $r_{ic}$

- Update model parameters: $\mu_c, \Sigma_c, \pi_c$

For each Gaussian (cluster):

- Update parameters

$$m_c = \sum_i r_{ic}; \pi_c = \frac{m_c}{m}$$

$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^i$$

$$\Sigma_c = \frac{1}{m_c} \sum_i r_{ic} \left(x^i - \mu_c\right)^T \left(x^i - \mu_c\right)$$

# Learning a GMM

Each EM step increases the accuracy of the model

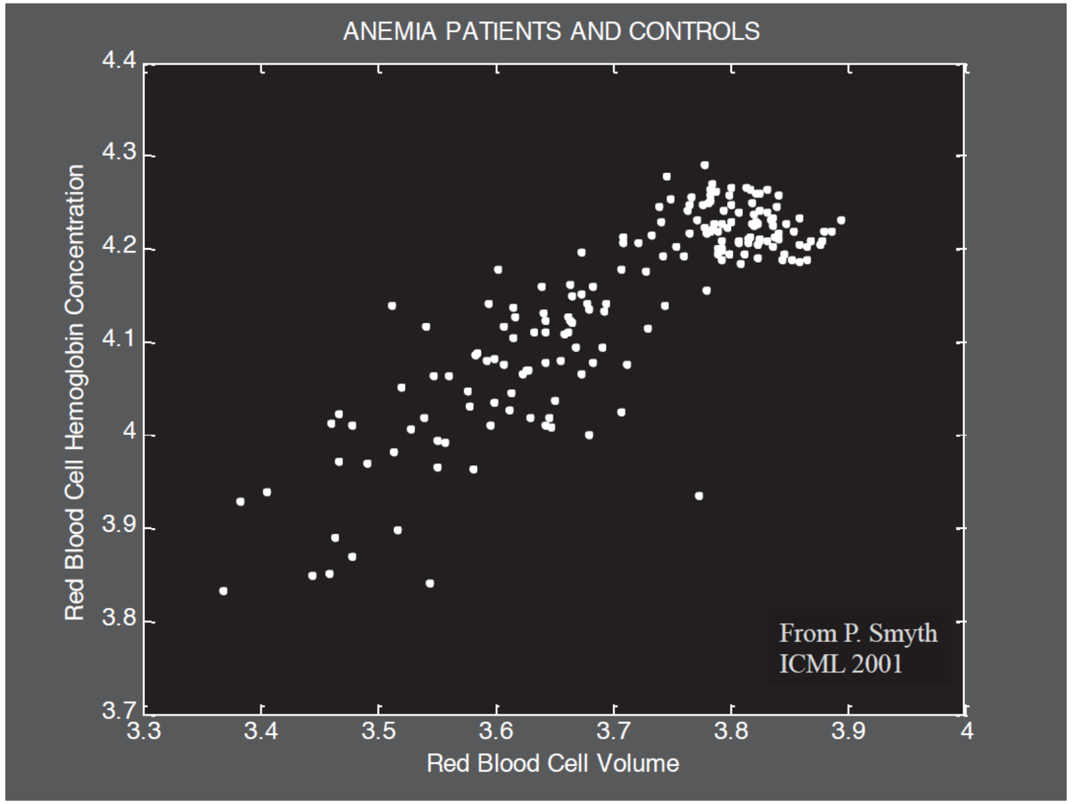
- Increases the log-likelihood

Iterate until convergence

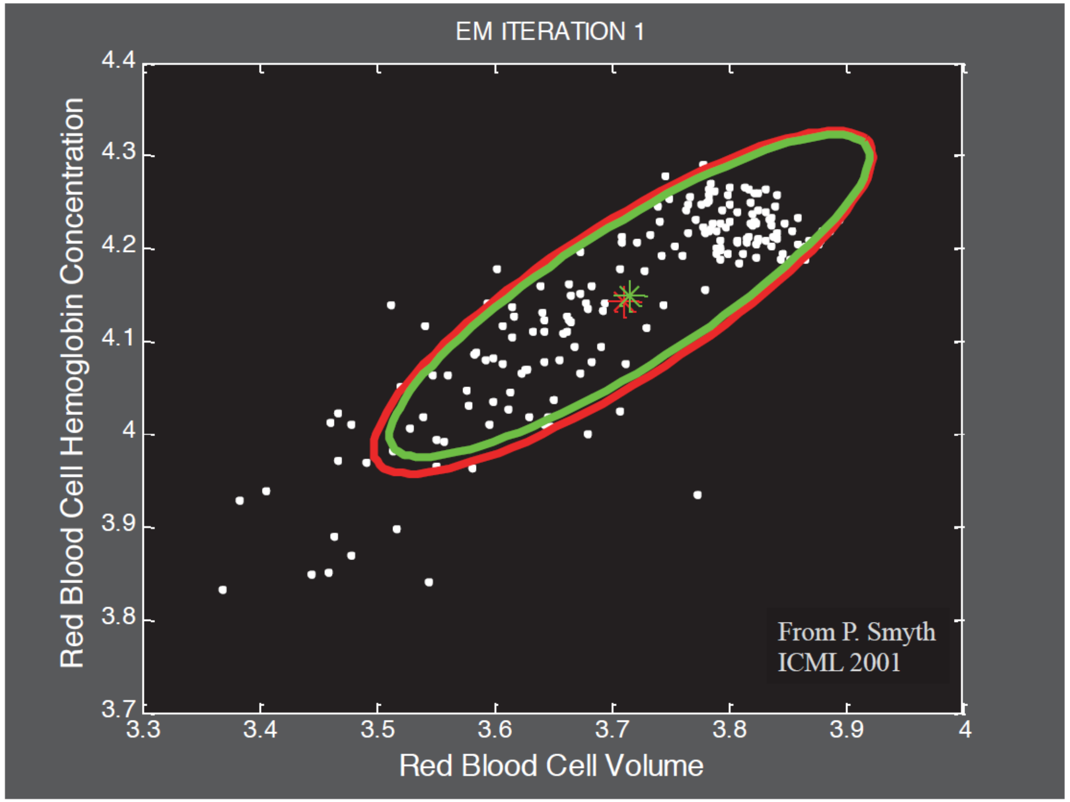
- It will converge

# EM - Example

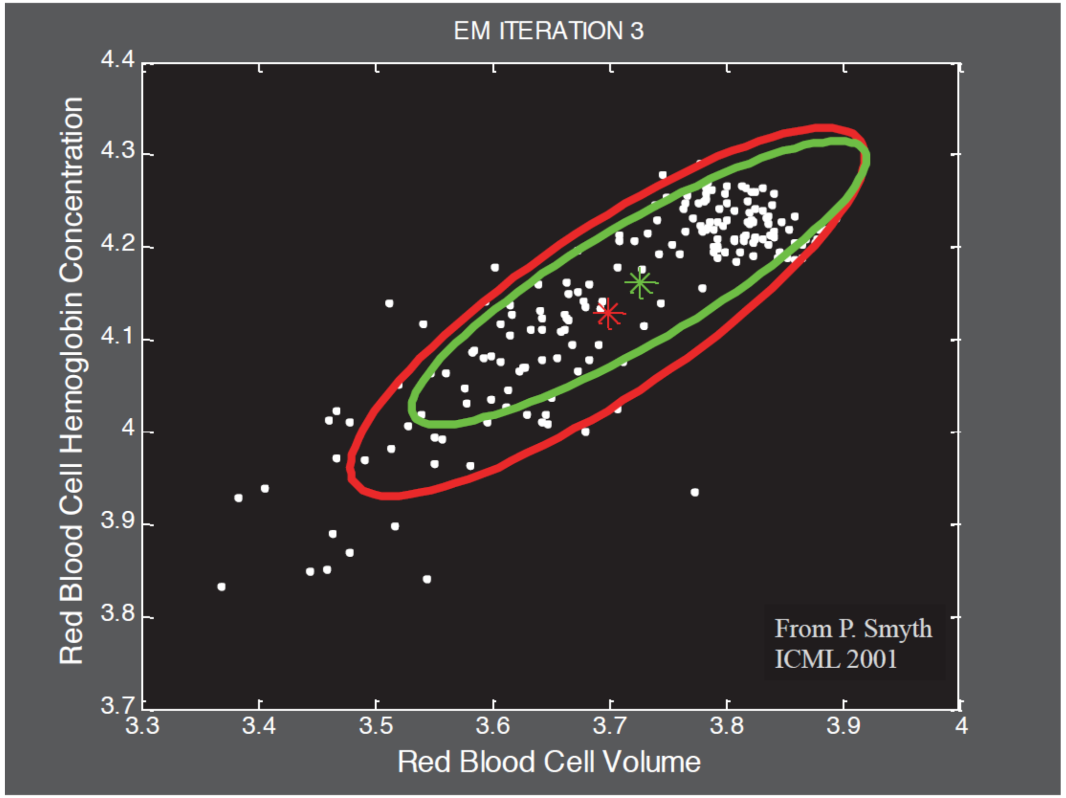
- Fit a GMM with two mixtures to this data



# EM - Example

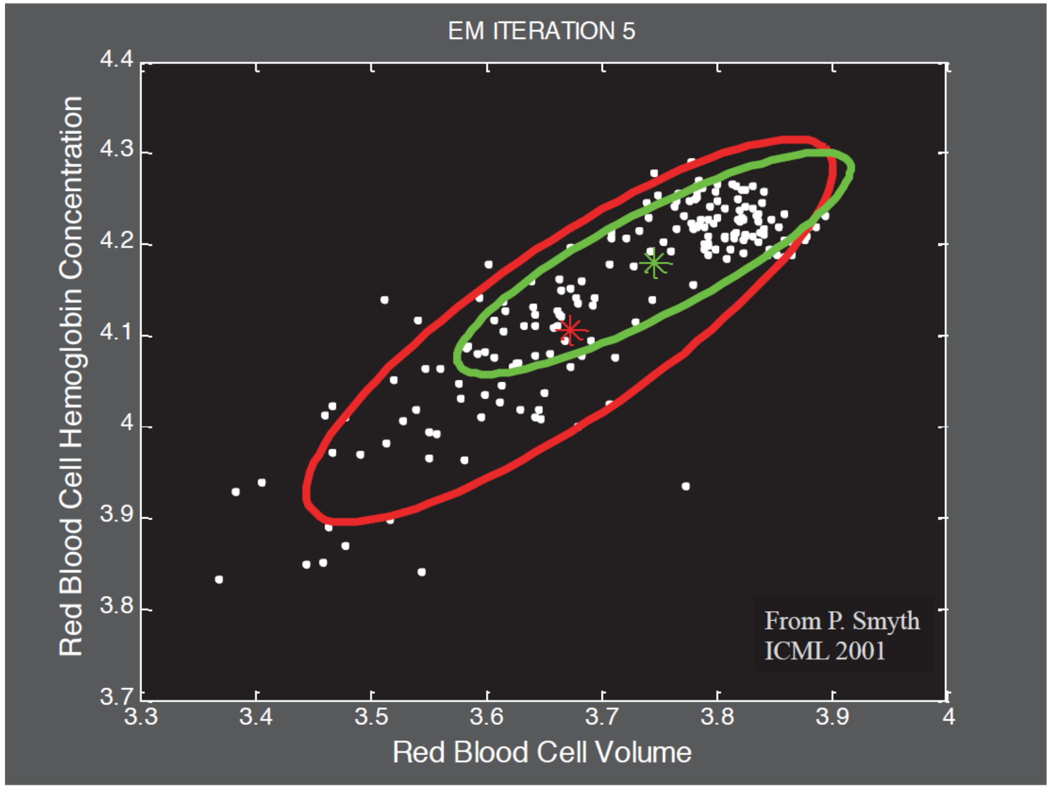

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

# EM - Example

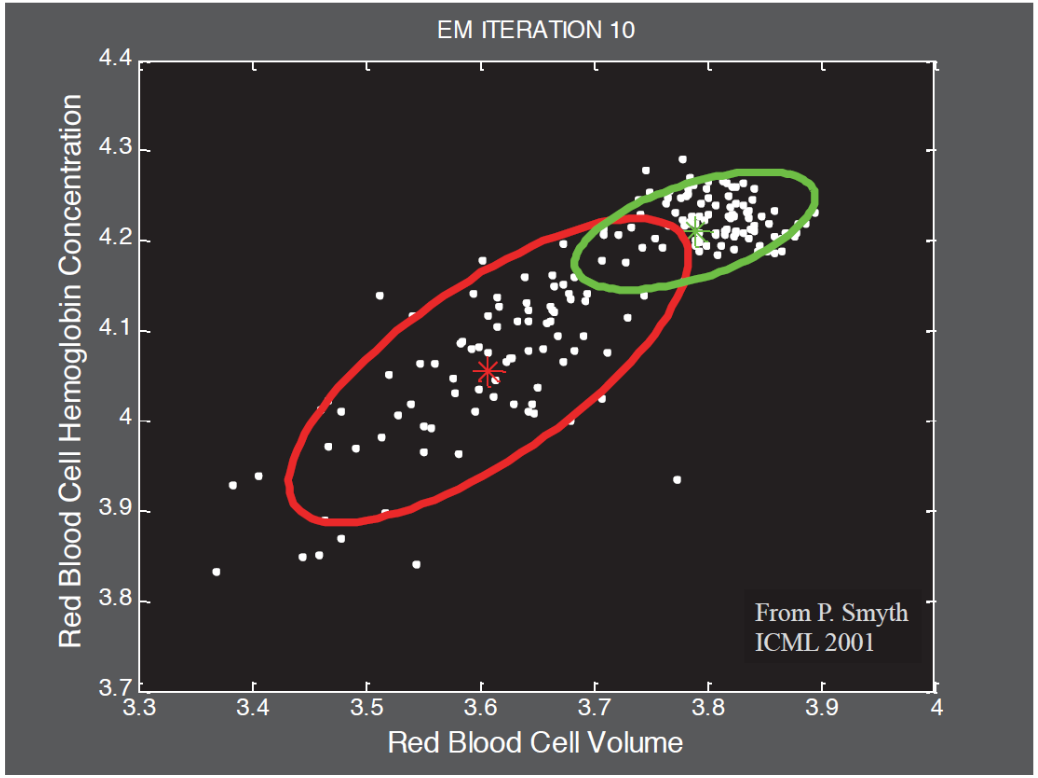


EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume
From P. Smyth
ICML 2001

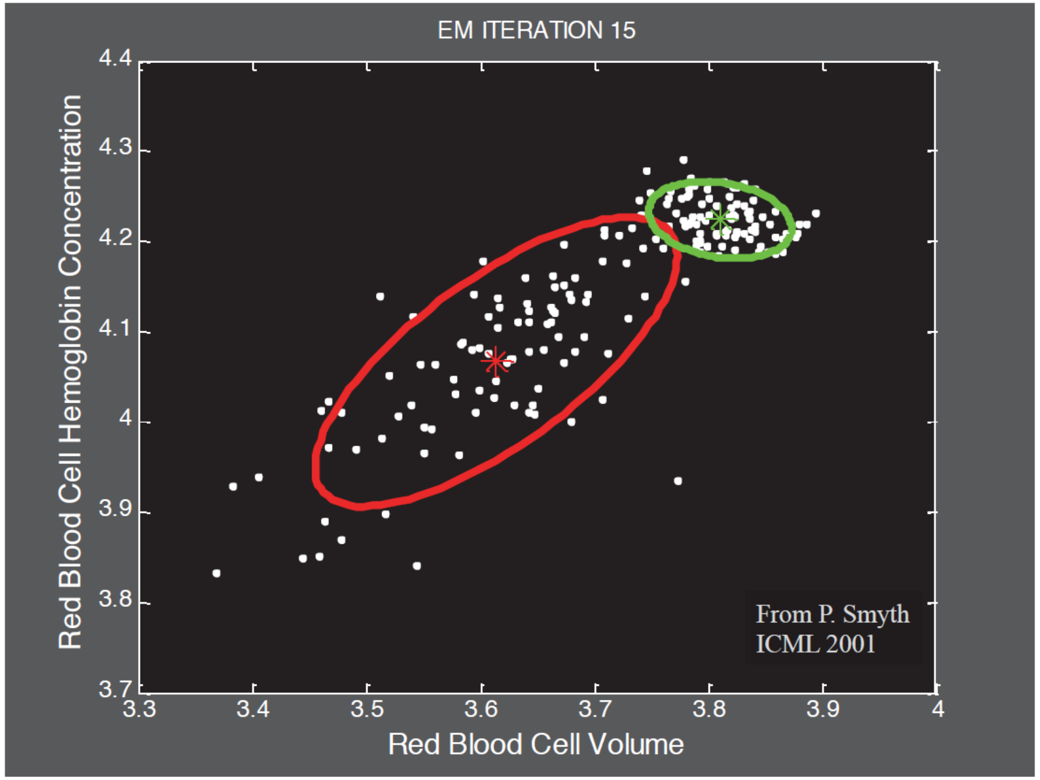
# EM - Example


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume
From P. Smyth
ICML 2001

# EM - Example

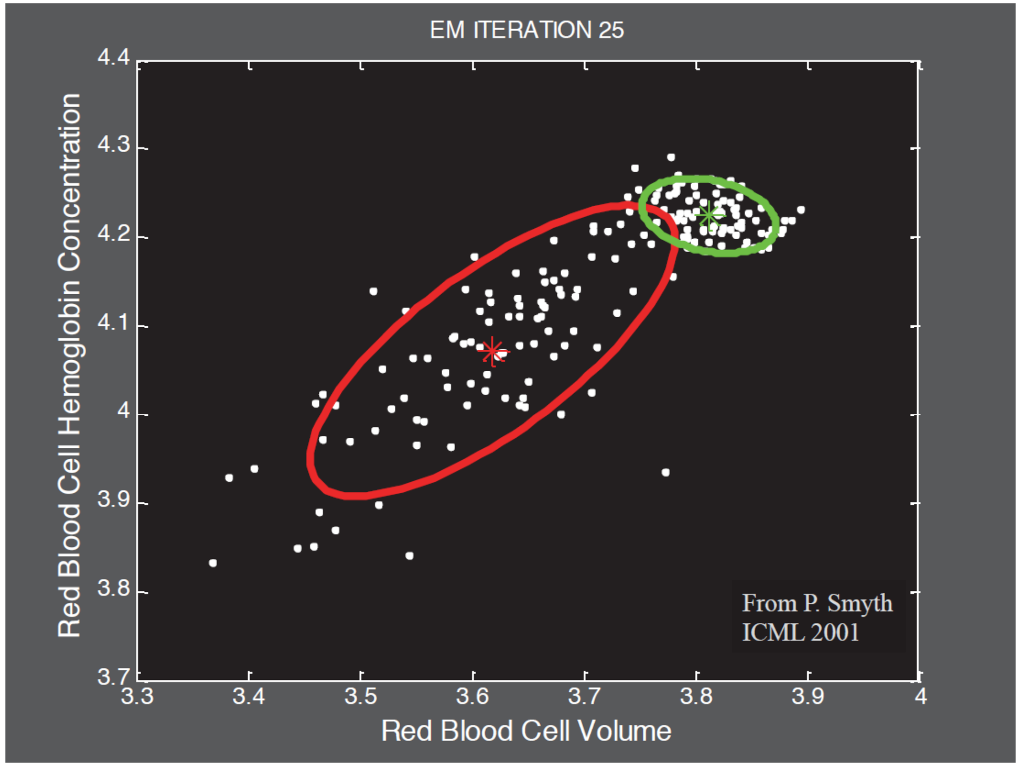

EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume
From P. Smyth
ICML 2001

# EM - Example



EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume
From P. Smyth
ICML 2001

# EM - Example

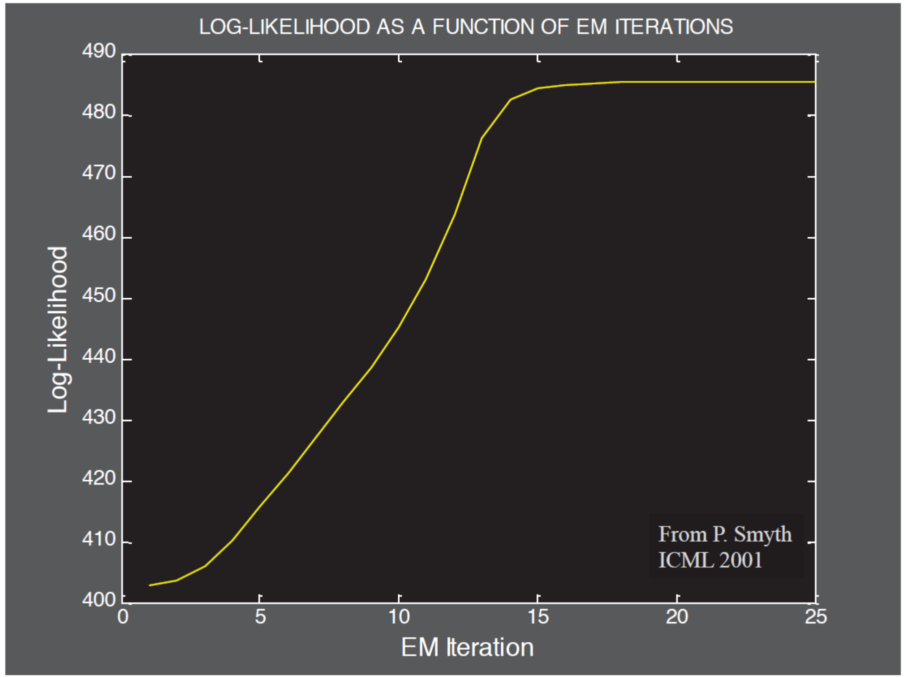


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume
From P. Smyth
ICML 2001

# EM - Example

Fit has converged

- Changes fast early
- Changes slower as the model nears the optimum

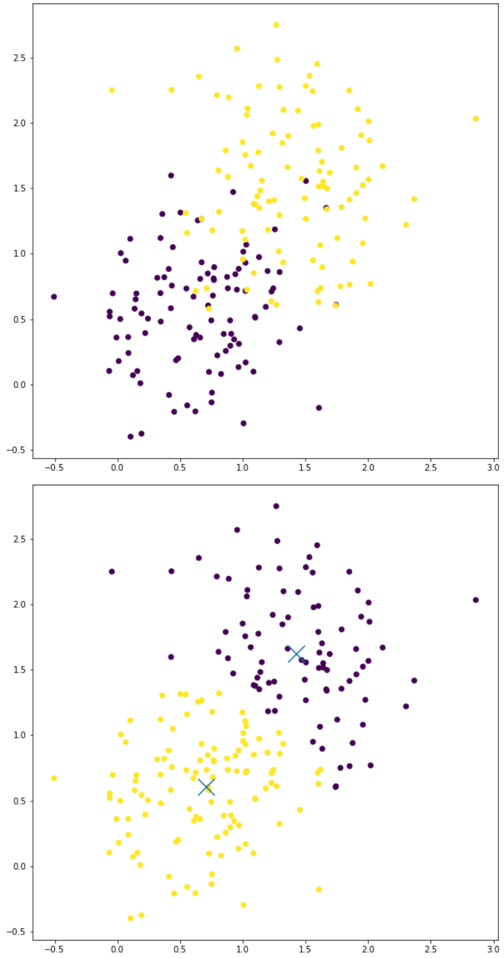


# An Example

- See ***CAB420_Clustering_Example_2_Gaussian_Mixture_Models.ipynb***
- Same data setup as K-Means example

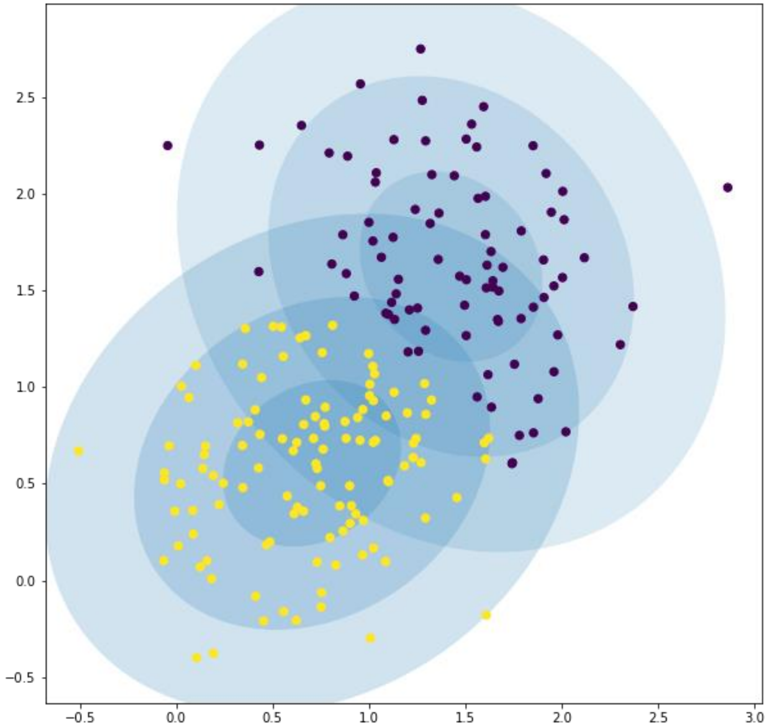
# Clustering Results

- Top: original data
- Bottom: GMM Results
- Cluster centres
  - (1.42573618, 1.62444572)
  - (0.70451705, 0.60575063)
- Cluster weights:
  - 0.43276892
  - 0.56723108
- Centres are close to true centres and weights are fairly even
  - This makes sense, we have 100 points in each true cluster



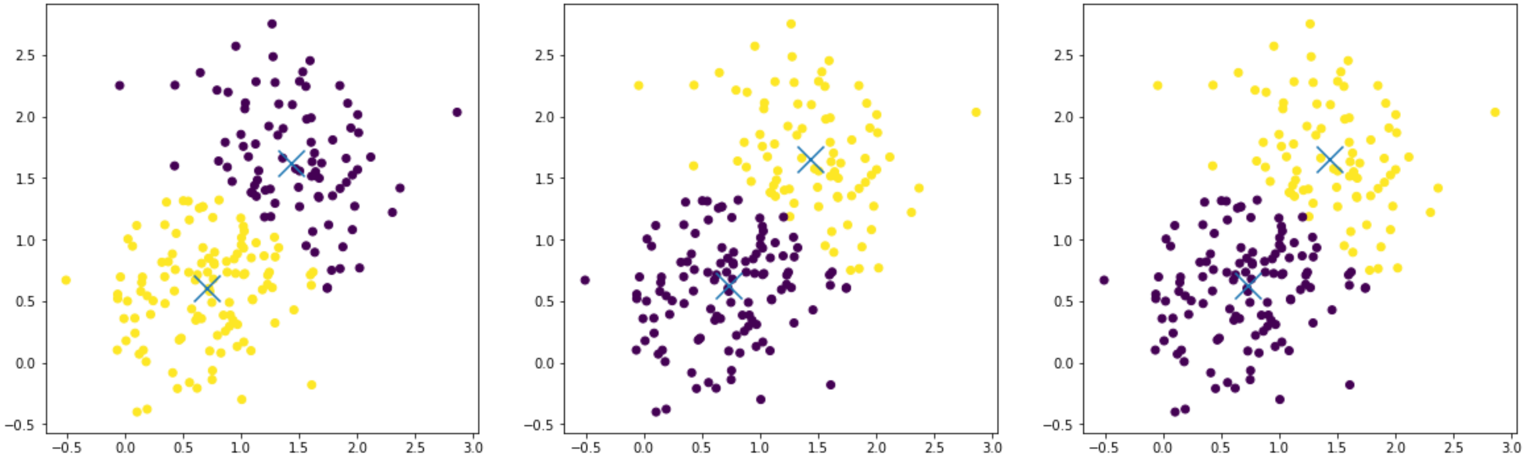
# GMMs and Soft Decisions

- GMMs don't have hard boundaries between clusters
  - Clusters can overlap
  - Points can belong partially to both clusters



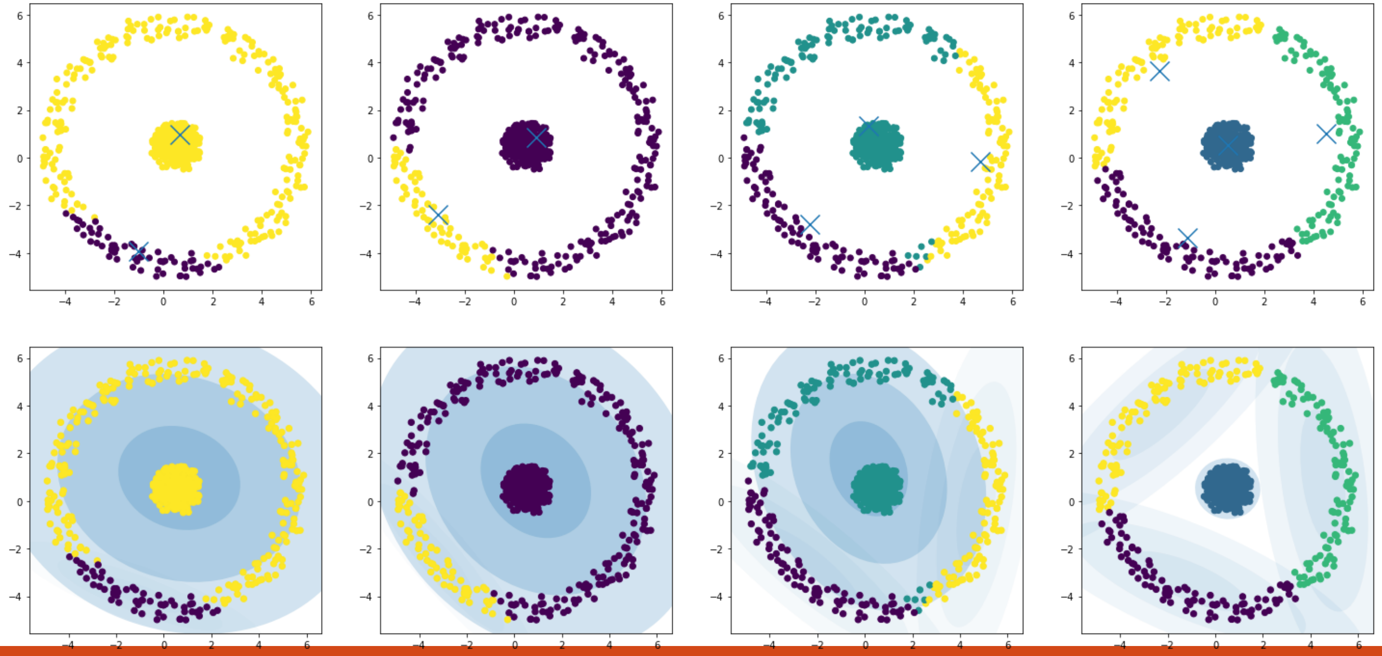
# GMMs and Randomness

- GMMs need an initial set of cluster centres
- Initial centres come from K-Means, which is impacted by randomness
  - Has a knock-on effect for the GMM
  - Variation typically less severe than with K-Means



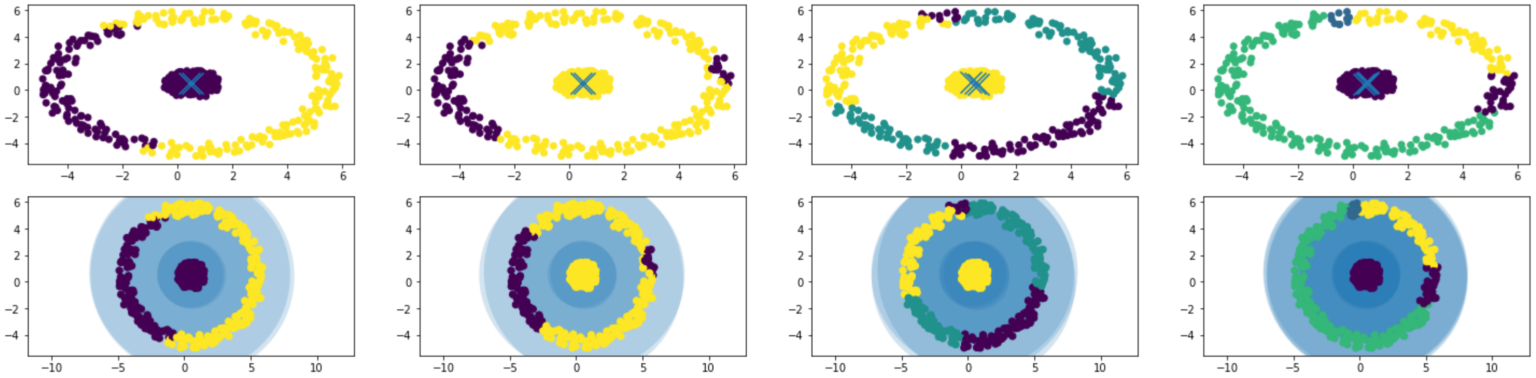
# A Second Test Case

- Overlapping clusters
  - Using K-Means for initialisation
  - Cannot separate data with K=2
  - Similar results to K-Means overall



# Random Initialisation

- We can initialise the GMM differently
  - Using other clustering results, our own estimates, or randomly
- We get several clusters with very similar centres
  - But different shapes, sizes and densities
- With careful initialisation, it would be possible to separate these clusters



# CAB420: How Many Clusters?

AND THE DARK ARTS OF MODEL SELECTION

# How do we select the number of clusters?

Is more clusters always better?

- Depends on our error measures

K-means cluster assignment cost

- $C\left(\underline{z}, \underline{\mu}\right) = \sum_i \left\|x_i - \mu_{ij}\right\|^2$
- Will decrease as clusters increase
  - More cluster centres, so on average points will be closer to a centre
- Need to add a penalty for model size

# Bayesian Information Criterion

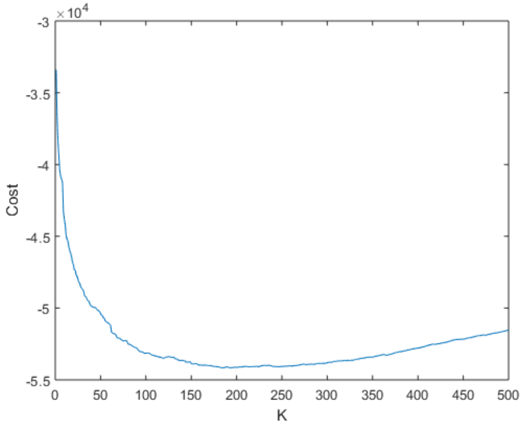
BIC

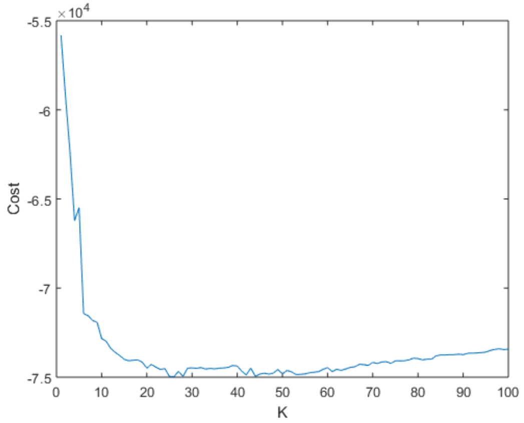
- Captures how informative a model is while also considering complexity
- Approximate form needed for K-means
- $J\left(\underline{z}, \underline{\mu}\right) = m \log\left(\frac{1}{m}\sum_i \left\|x_i - \mu_{ij}\right\|^2\right) + k \log m$
  - $m$ = number of samples
  - $k$ = size of the model (number of parameters)
- $k \log m$ will increase with model complexity, first term must decrease by enough to make the extra parameters “worth it”

# Optimum Number of Clusters

Not the same for K-means and a GMM. Why?

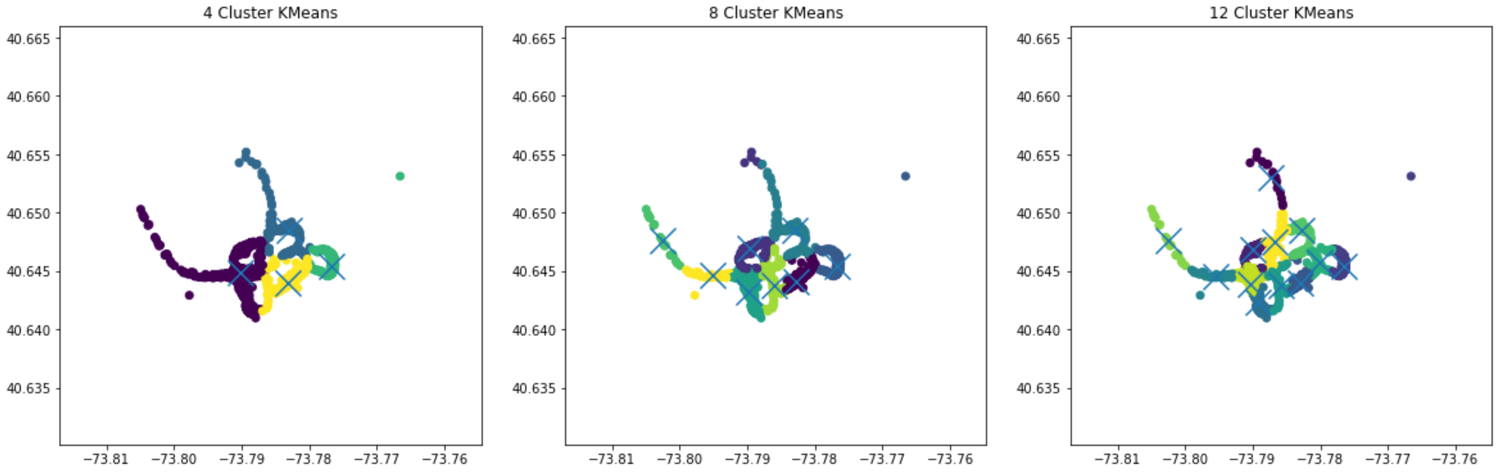
- Different number of parameters
  - GMM has many more parameters
- Approximations in K-means BIC formulation
  - Impacts accuracy



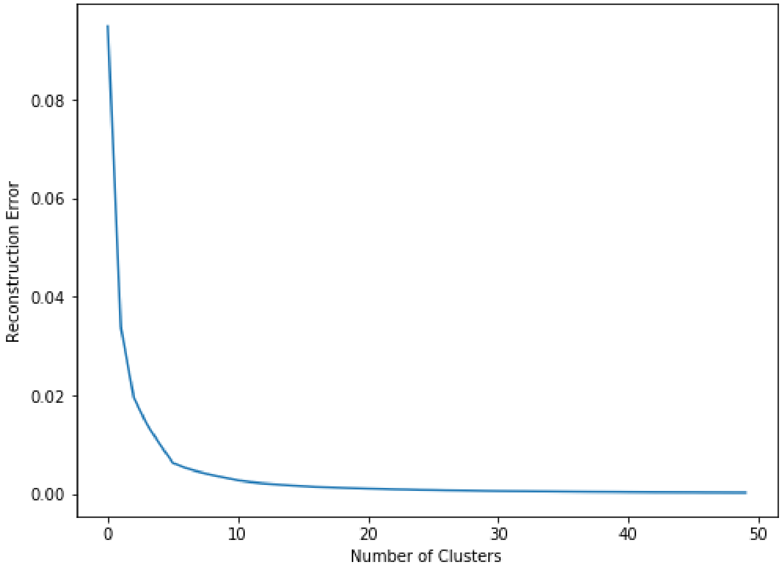
# An Example

- See ***CAB420_Clustering_Example_3_How_Many_Clusters.ipynb***
- Our data
  - New York taxi data
  - Focus on drop-off locations around JFK airport



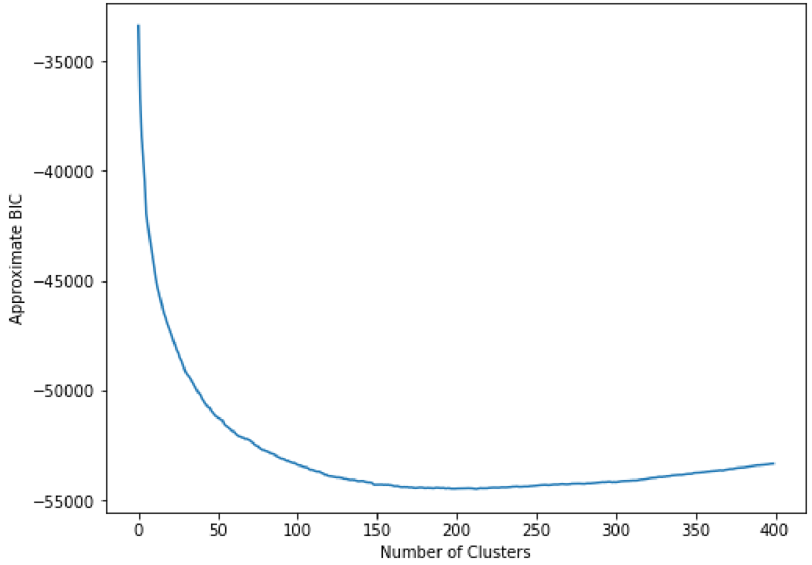
# K-Means: Selection of K

- Reconstruction error
  - Average distance to assigned cluster centre
- As K increases, error decreases
- Can use the "elbow" point of the curve as a metric to choose K
  - ~5 in this case
  - Very much a heursistic, but not a bad one all the same



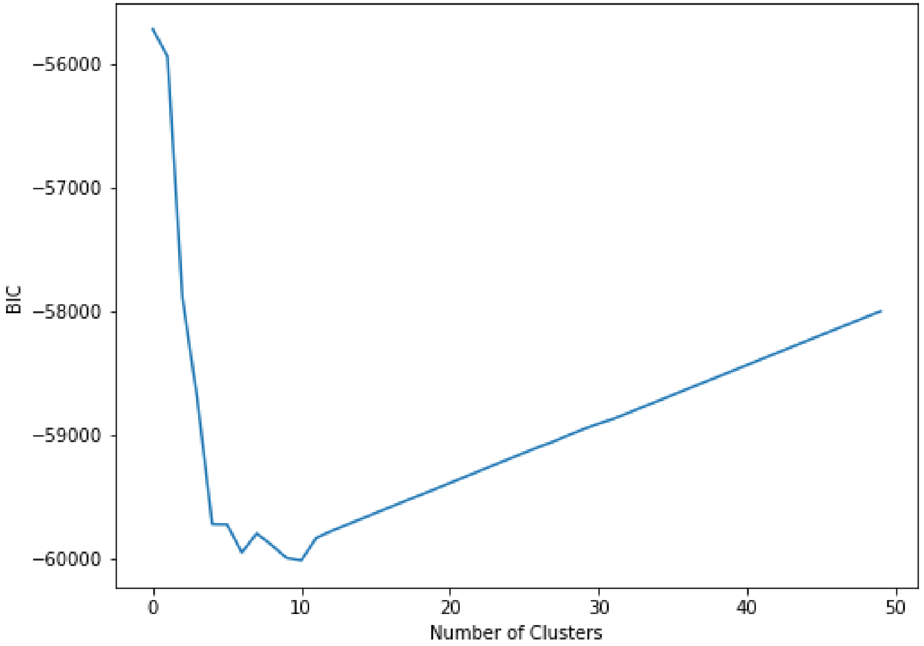
# K-Means: Selection of K

- Approximate BIC
  - Reconstruction error plus a term for model complexity
- Minimum of curve is best value of K
  - ~200 in this case
  - Approach somewhat sensitive to scale of data (this impacts reconstuction error)
- Very different value of K to before



# GMM: Selection of K

- BIC
  - Combination of model complexity and error
- Minimum of curve is best value of K
  - ~10



# Why is K different each time?

- K-Means vs GMMs
  - GMMs have more parameters, so complexity penalties are larger for the same K
- Reconstruction cost is dependent on data scale
  - Data that has a very small range will have smaller reconstruction costs
  - Can lead to big differences between looking at reconstruction curve "elbow" and approximate BIC minimum

# Selecting K

- Methods offer a suggested value
  - There may be reasons to use a different value
- Judgement of problem/data/solution requirements is important
  - Does a high K make analysing results too hard?
  - Does a small K risk grouping things together that are (or should be) distinct?
- You may have prior knowledge to help inform selection of K
  - You may know that there are 10 actual things to cluster
  - If you have prior knowledge, use it
- There are other methods to select K
  - Silhouette score for example

# What happens with K is wrong?

- Two possible errors:
  - Over-clustering
    - True clusters are split into multiple sub-clusters, i.e. we have too many clusters
  - Under-clustering
    - True clusters are merged into a single cluster, i.e. not enough clusters

- Hard to work out what’s happening when the true clusters aren’t known.

# CAB420: Clustering Applications

CLUSTERING ACTUAL DATA

# Knowledge Discovery

- Given a dataset, try to extract some useful information to make sense of the data
  - Very broad and vague, what is "useful"?
- Clustering is one approach to help
  - Identify a small set of typical samples
    - Cluster centres
    - Clusters may have semantic meanings
  - Determine distribution of samples based on clustering
    - Which clusters are most common?
    - Do cluster occurance rates change over time?

# An Example

- See ***CAB420_Clustering_Example_4_Clustering Applications.ipynb***

- The Data
  - Bike share data from NY
  - Three months: July and December 2019, and July 2020

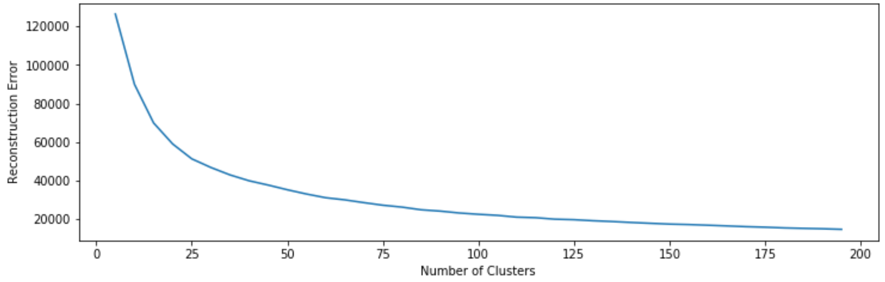
- Our Task
  - Compare usage between the three months

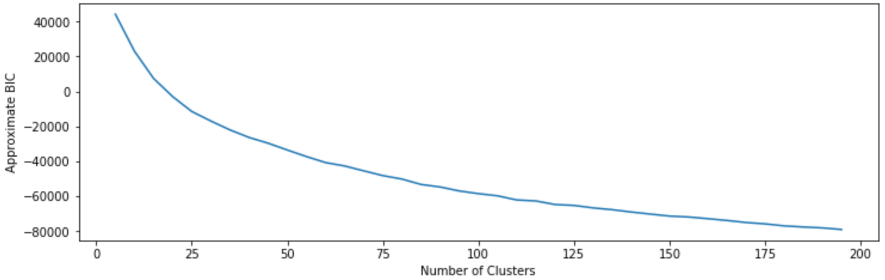
# Data setup and pre-processing

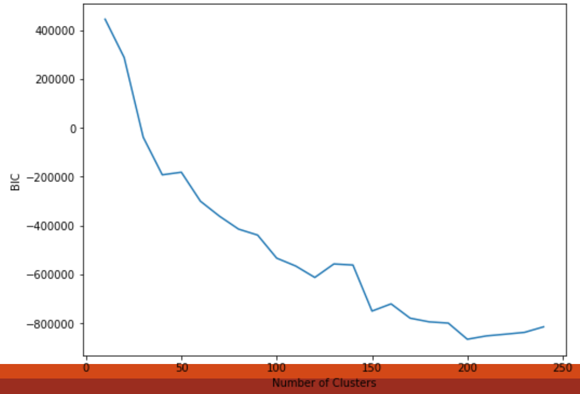
- Use five dimensions
  - Trip duration (in seconds)
  - Start location (lat, lon)
  - End location (lat, lon)
- Dimensions have very different scales
  - Standardise data
- Some trips are very long (days or more)
  - Remove trips over 2 hours in length
    - Somewhat arbitrary choice

# Selecting K

- K-Means (top)
  - Elbow of reconstruction curve (left) is at ~20
  - Minimum of approximate BIC is >200
- GMM (bottom)
  - Minimum of BIC at ~200



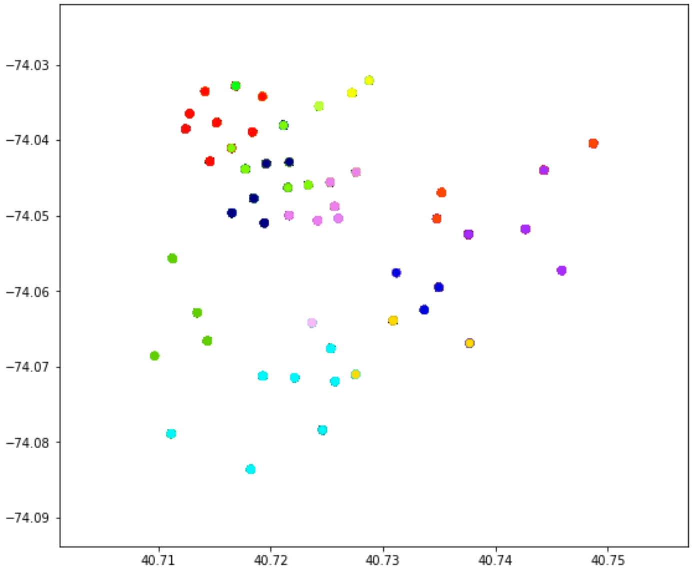



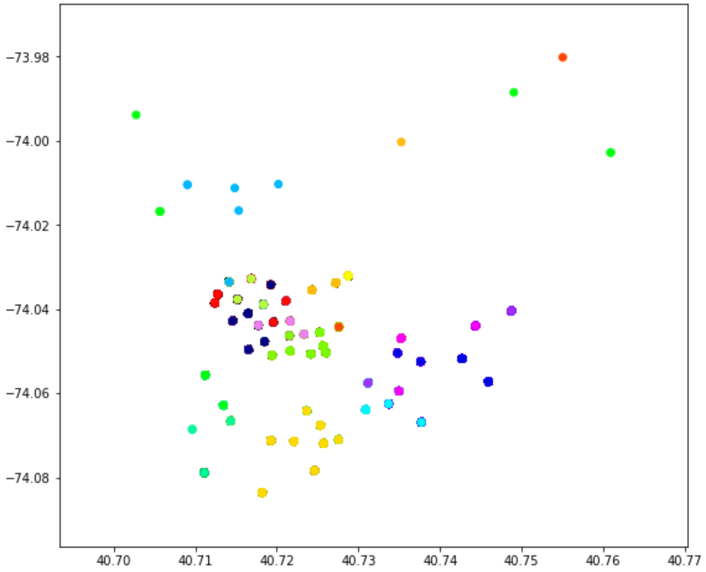
# Selecting K

- Very different values of K for each plot
- Consider the application and data
  - Knowledge Discovery: we're seeking to use clustering to find a set of representative points (bike trips) which we can use to analyse the data
  - How many "types" of trip are possible?
    - How many can we reasonably analyse/compare?
  - Can we assign semantic meaning to clusters?
    - If so, how many and at what granularity?
- We'll stick to a smaller value of K to simplify analysis
  - K=20
  - Large enough to group very different behaviours
  - Small enough to keep analysis simple
  - Likely leading to under-clustering, in that true clusters are being merged

# K-Means Clusters

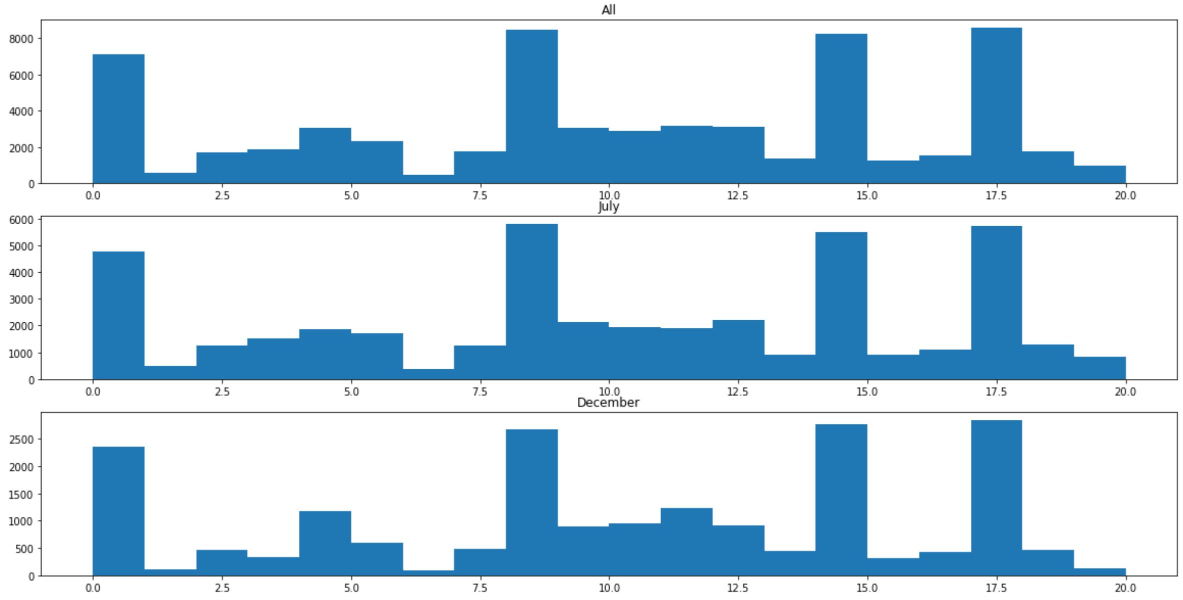
- Start location (left) and end location (right) shown
  - Trip Duration not shown
  - Hard to plot 5D data
- Inspection of cluster centres (see example) shows that a couple of clusters capture long trips

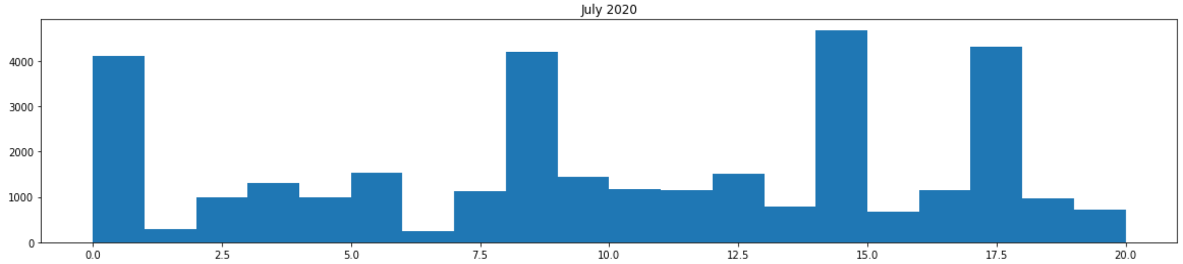


# K-Means Analysis

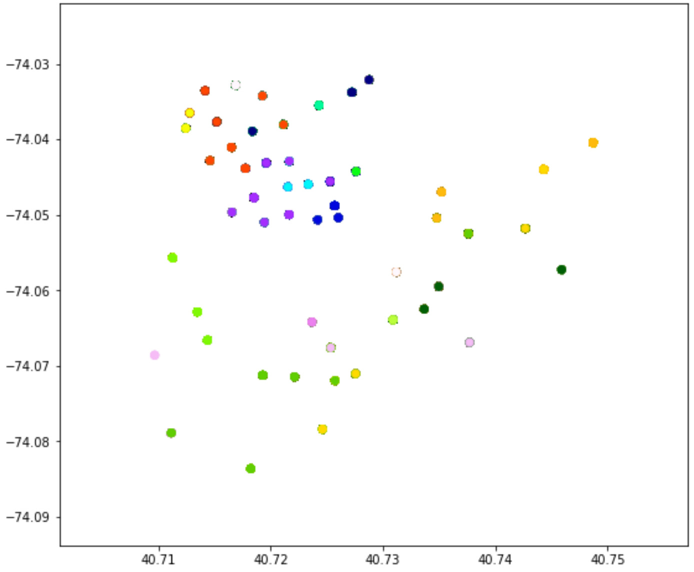
- Patterns of use visualised by looking at how often points in each cluster occur
- Overall patterns similar across all considered time periods

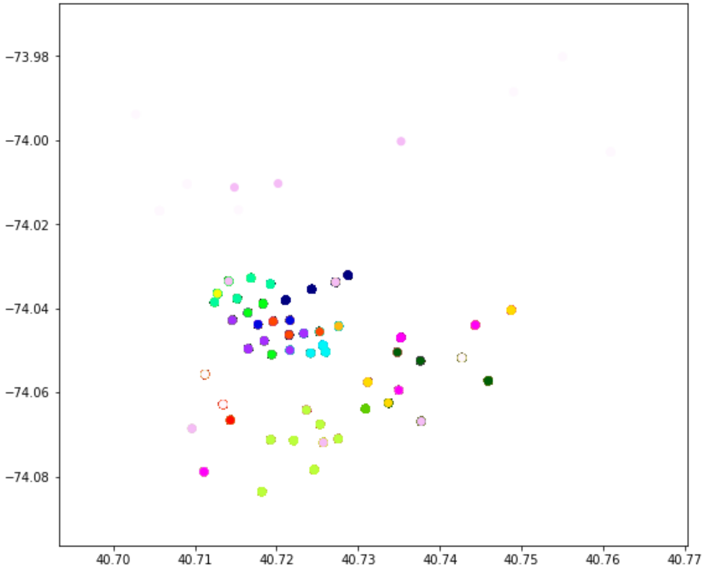


# GMM Clusters

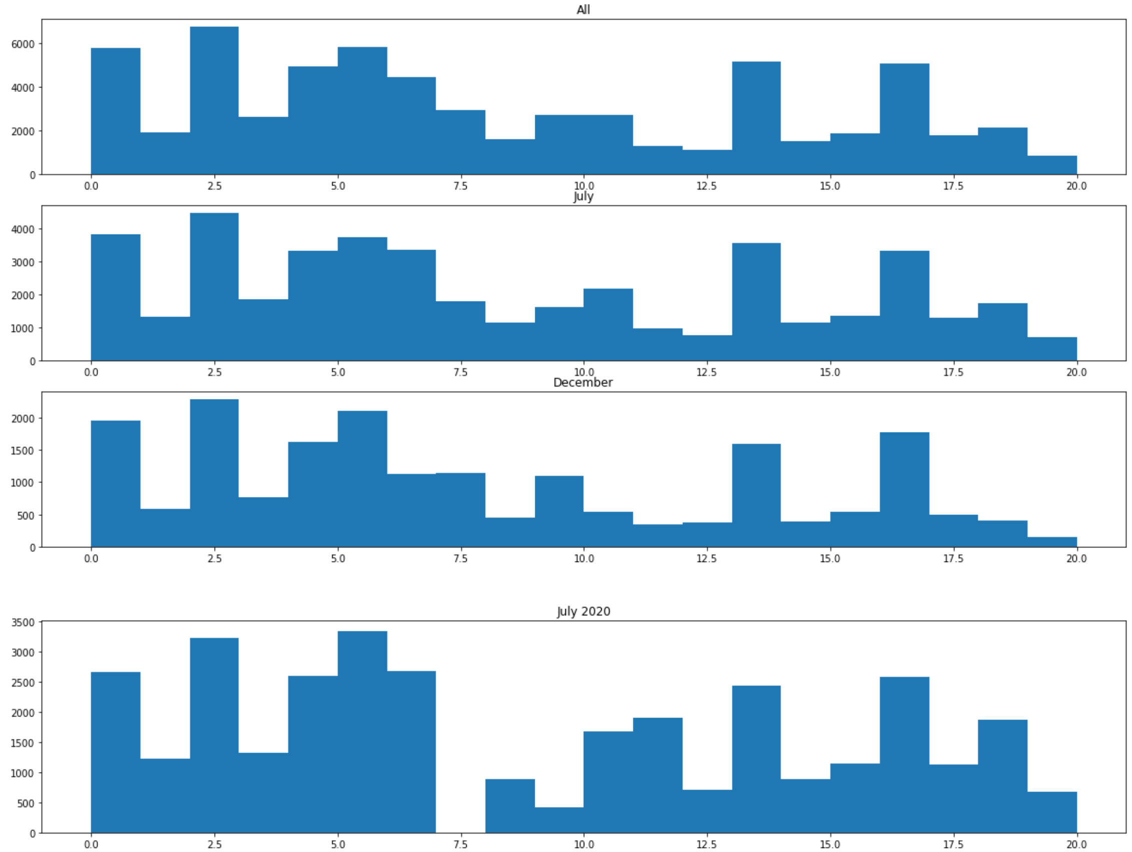
- Similar to K-Means, but clusters less uniform in size
  - GMMs allow cluster size and shape to vary
  - K-means limited to spherical clusters



# GMM Analysis

- Same visualisation as K-Means
- Both 2019 time periods appear very similar
- More pronounced differences between 2019 and 2020



# Further Thoughts

- Distance metrics are important
  - Is the way points are being compared valid?
  - We have locations (lat, lon) and durations? Is Euclidean distance appropriate?
- Cluster distributions can be compared numerically
  - Histogram intersection, Bhattacharya distance
    - See example
- Number of clusters will impact analysis
  - More clusters will tend to better highlight differences
  - Too many clusters will show differences where they're actually aren't any

# Anomaly Detection

- Given a set of data, find points that are unusual or abnormal
  - Also referred to as outlier detection
- Typical approach is
  - Train a model on normal data
  - Evaluate the model on a new set of data
  - Any point that is a sufficiently poor fit to the data is an outlier
    - Requires a threshold to define what "sufficiently poor" is
- Value of K can impact performance
  - Larger K means more clusters, which means points overall will fit the model better

# An Example

- See ***CAB420_Clustering_Example_4_Clustering Applications.ipynb***

- The Data
  - Same as before (Bike share data from NY)
  - July and December 2019 from the training set
  - July 2020 is the test set

- Our Task
  - Find abnormal trips in the test set

# Data setup and pre-processing

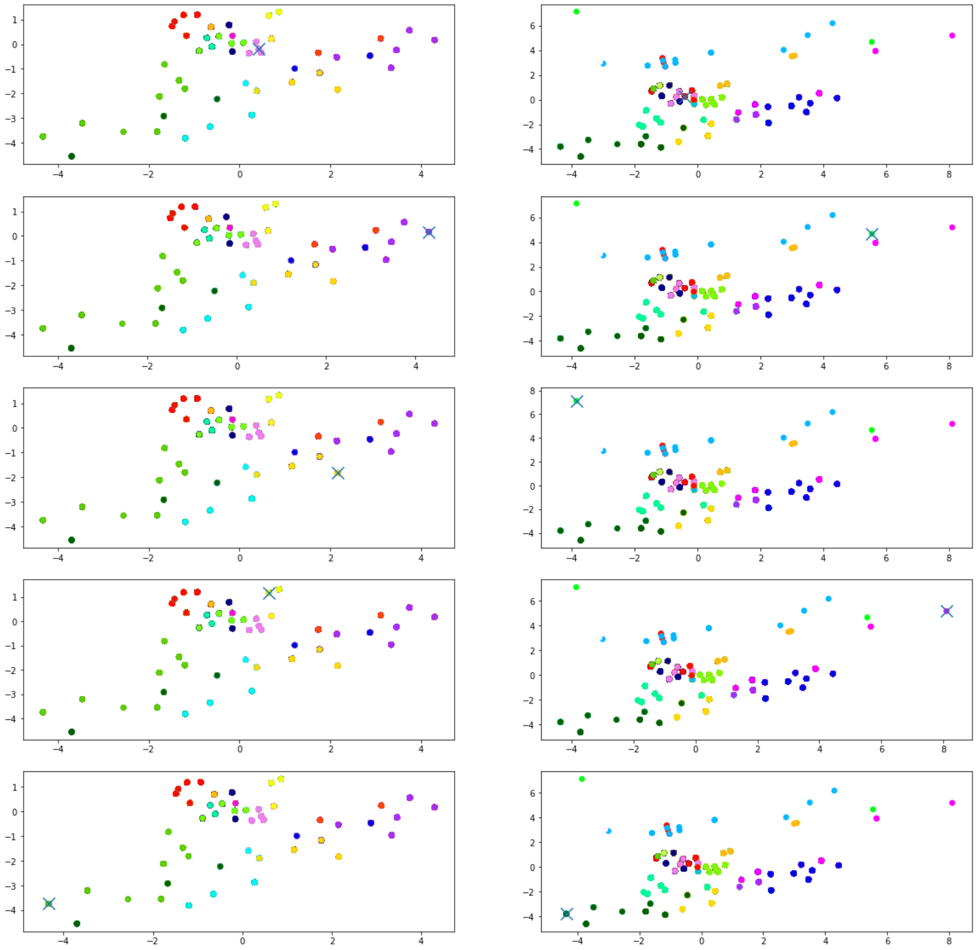
- Same as for previous application
  - Same dimensions
  - Use the same value of K for clustering approaches
- Only concerned with anomalies in July 2020
- Rather than set a threshold, we'll find the set of the most abnormal points
  - Ideally, to set a threshold we'd have a dataset with known anomalies, and use this to tune a threshold to reach the desired detection sensitivity

# Anomalies and K-Means

- We can use distance to assigned cluster centre as a proxy for how unusual a point is
  - Limited in that it can identify points that lie at the boundary of two clusters
  - Can be misleading as it does not consider cluster spread or density

# Anomalies and K-Means

- Abnormal points are dominated by long trips
  - Longer trips lead to a larger distance, even with standardised data



# Abnormalities and GMMs

- GMMs allow us to determine the likelihood of a point
  - How likely is it that this point belongs to this distribution?
- Allows us to identify highly unlikely (abnormal) points

# Abnormalities and GMMs

- Abnormal trips are a mix of long and short trips
  - Seems more realistic than the K-Means results
- All abnormal points belong to cluster 19
  - Seems odd, suggests that the clustering results need further investigation
  - Could be under-clustering and have several behaviours grouped together