AI-503 Advanced Machine Learning

Recap

- Introduction to PCA
- Why PCA?
- Mathematical Formulation
- Interpretation

PCA

- If we have very high dimensional data, we can reduce its dimensionality by projecting it along directions (or vectors) such that the variance along the chosen direction is maximized in order to preserve the most information in the data.

- Finding the direction of maximum variance for a given data set corresponds to **<u>finding the eigen vector of the covariance matrix</u>**
- Find the axes, over which if I project the data, the variance is maximized
- Project the data over the axes preserving the highest amount of variance.

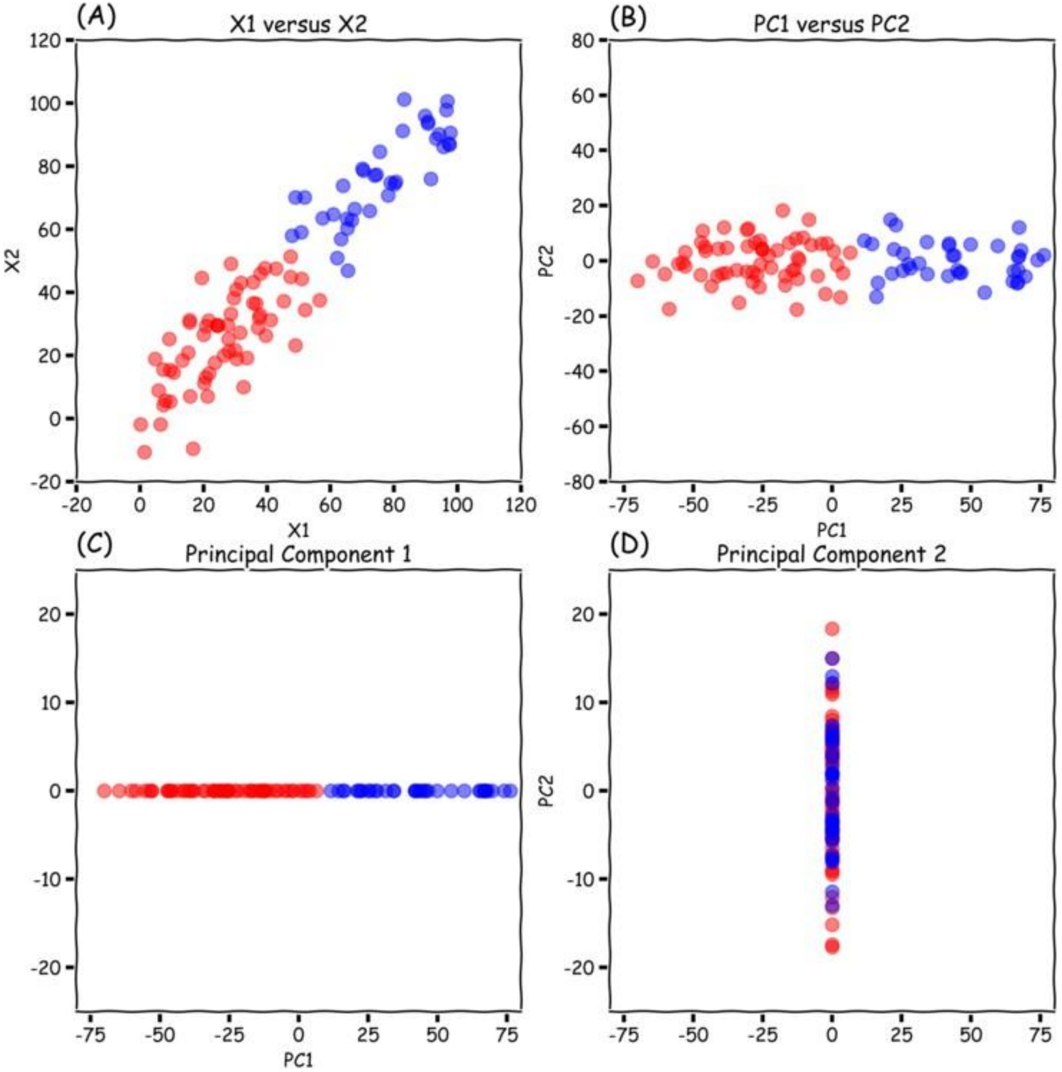
PCA

- 1. Find the principal components
- 2. Transform data by projecting over principal components
- But where is the dimensionality reduction?
  - Select a smaller number of principal components for transformation than the actual dimensionality
- Implemented in Sklearn
  - https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- Demo: https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.09-Principal-Component-Analysis.ipynb#scrollTo=6tCGHfiDfmeN

PCA- Limitations

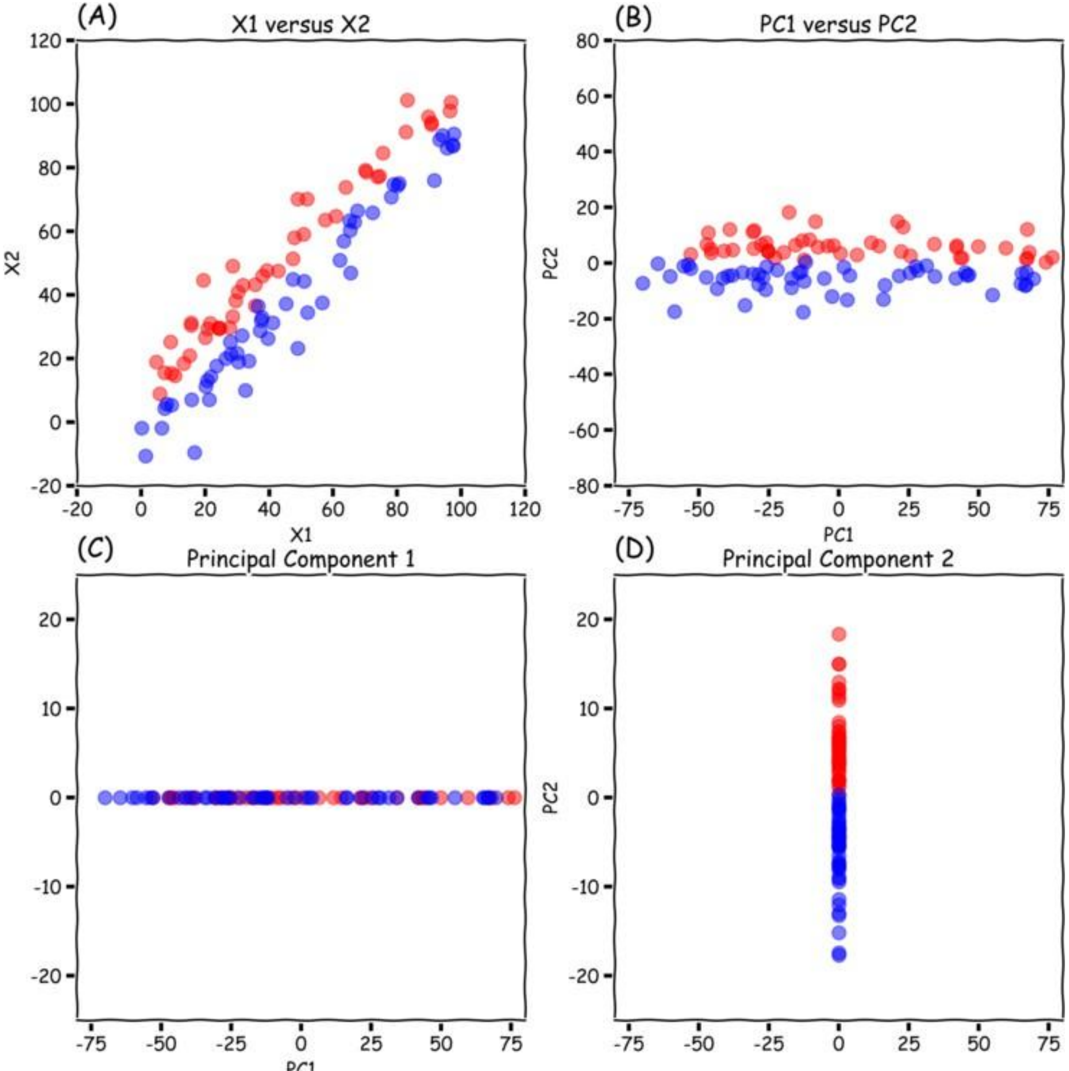
- Do not use PCA if you want to retain the original features.
  - PCA transforms data and new features are created.

PCA Dim. Reduction for classification



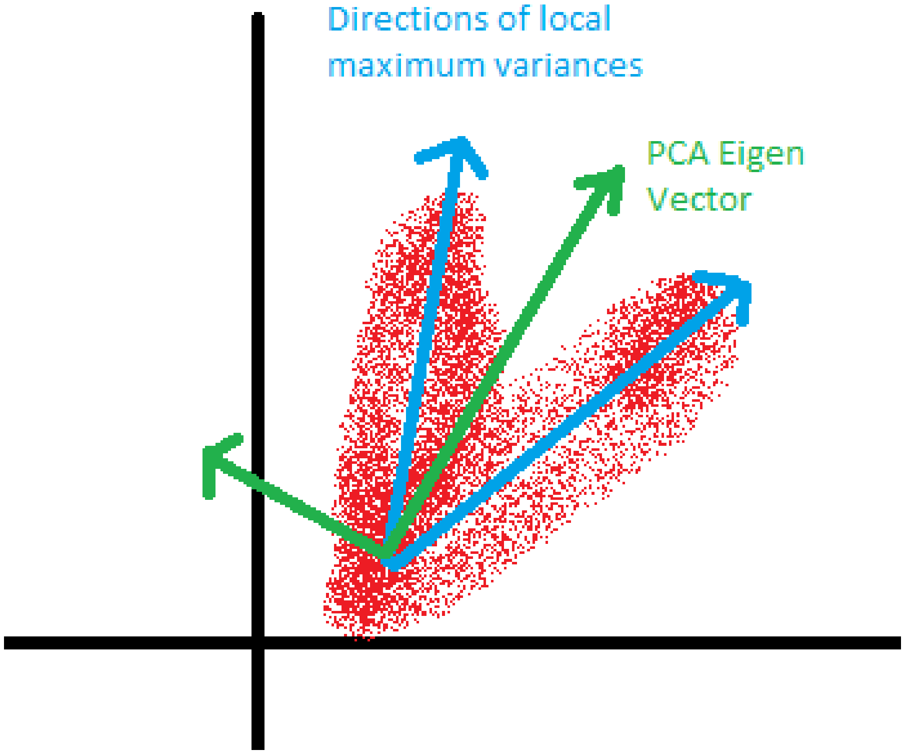
PCA- Limitations

- PCA reduction may make a previously separable classification problem inseparable



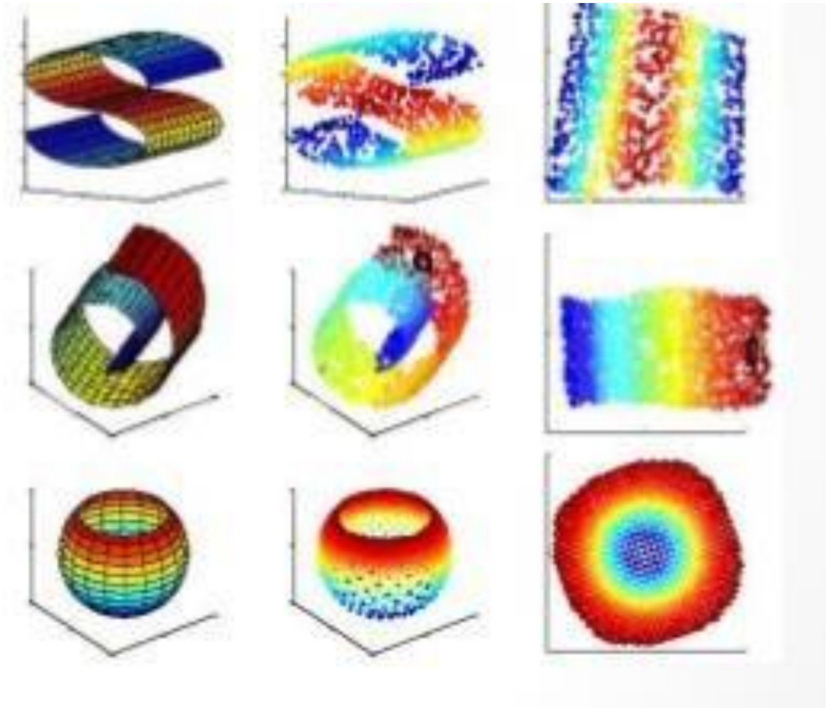
PCA-Limitations

- PCA projects data over orthogonal principal components
- Sometimes, data may demand non-orthogonal principal components to represent the data



PCA-Limitations

- Assumes that the subspace on which data “lives” is linear. What if the surface important to classification is non-linear?
- Dimensionality reduction using PCA would cause significant loss in information



So to perform non-linear dimensionality reduction

- Kernelized PCA
- Locally linear methods
- Manifold learning methods
- https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction

PCA in Python

```
>>> import numpy as np
 >>> from sklearn.decomposition import PCA
>>> pca = PCA(n_components=2)
>>> pca.fit(X)
PCA(n_components=2)
 >>> (pca.explained_variance_ratio_)
 [0.9924... 0.0075...]
>>>(pca.singular_values_)
[6.30061... 0.54980...]
>>> pca.transform(X)
```

Required reading

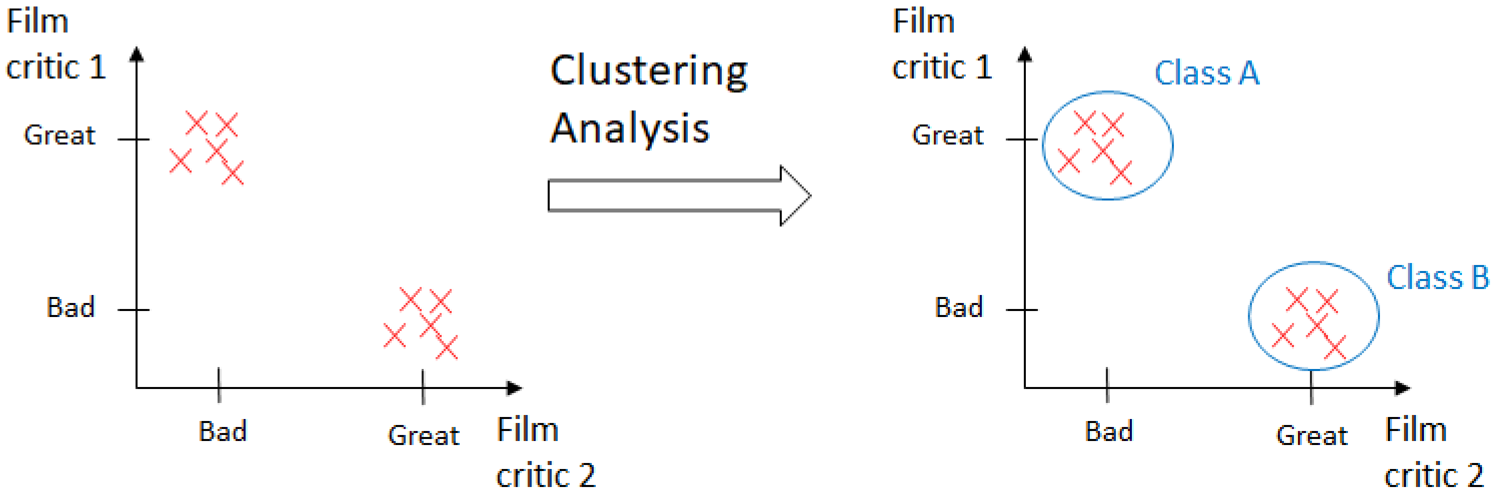
- Please go through the following tutorial
- https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Clustering

- A way of grouping together data samples that are ***similar*** in some way - according to some criteria that you pick
- A form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together
- So, it's a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest

Clustering

- to find different groups within the data.
  - find the structure in the data so that elements of the same cluster (or group) are more similar to each other than to those from different clusters



https://towardsdatascience.com/unsupervised-machine-learning-clustering-analysis-d40f2b34ae7e

Applications

- Image processing
- Biology
  - taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval
  - document clustering
- Land use
  - Identification of areas of similar land use in an earth observation database
- Marketing
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning
  - Identifying groups of houses according to their house type, value, and geographical location
- Climate
  - understanding earth climate, find patterns of atmospheric and ocean
- …

Clustering Approaches

- Partitioning based approaches
  - Construct various partitions and then evaluate them by some criterion
  - K-means, k-medoids, CLARANS
- Hierarchical methods
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Agglomerative and Divisive
  - Diana, Agnes, BIRCH, CAMELEON
- Density based methods
  - Based on connectivity and density functions
  - DBSACN, OPTICS, DenClue

Similarity

- to find different groups within the data.
  - find the structure in the data so that elements of the same cluster (or group) are **more similar** to each other than to those from different clusters
- How do you define similarity among data points?

Similarity

- Recall that the goal is to group together "similar" data – but what does this mean?
- No single answer – **it depends on what we want to find or emphasize in the data**
- The similarity measure is often more important than the clustering algorithm used – don't overlook this choice!

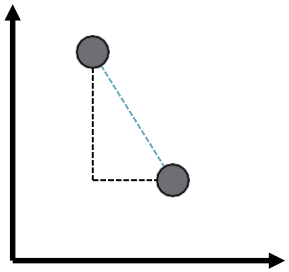
(Dis)similarity

- Instead of talking about similarity measures, we often equivalently refer to dissimilarity

- A **dissimilarity** measure is a function f(**x**,**y**) such that f(**x**,**y**) > f(**w**,**z**) if and only if **x** is less similar to **y** than **w** is to **z**

- This is always a *pair-wise* measure

Euclidean distance



$$d_{euc}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Here *n* is the number of dimensions in the data vector.

Pearson Linear Correlation

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - \left(\sum x_i\right)^2}\,\sqrt{n\sum y_i^2 - \left(\sum y_i\right)^2}},$$

Pearson Linear Correlation

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the values
- Always between –1 and +1 (perfectly anti-correlated and perfectly correlated)
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

Pearson Linear Correlation

- Pearson linear correlation (PLC) is a measure that is invariant to scaling and shifting (vertically) of the values
- Always between –1 and +1 (perfectly anti-correlated and perfectly correlated)
- This is a similarity measure, but we can easily make it into a dissimilarity measure:

Partition based algorithms

- Partitioning method: Construct a partition of n examples into a set of K clusters
- Given: a set of examples and the number K
- Find: a partition of K clusters that optimizes the chosen partitioning criterion
  - Globally optimal
    - Intractable for many objective functions
  - Effective heuristic methods: K-means and K-medoids algorithms

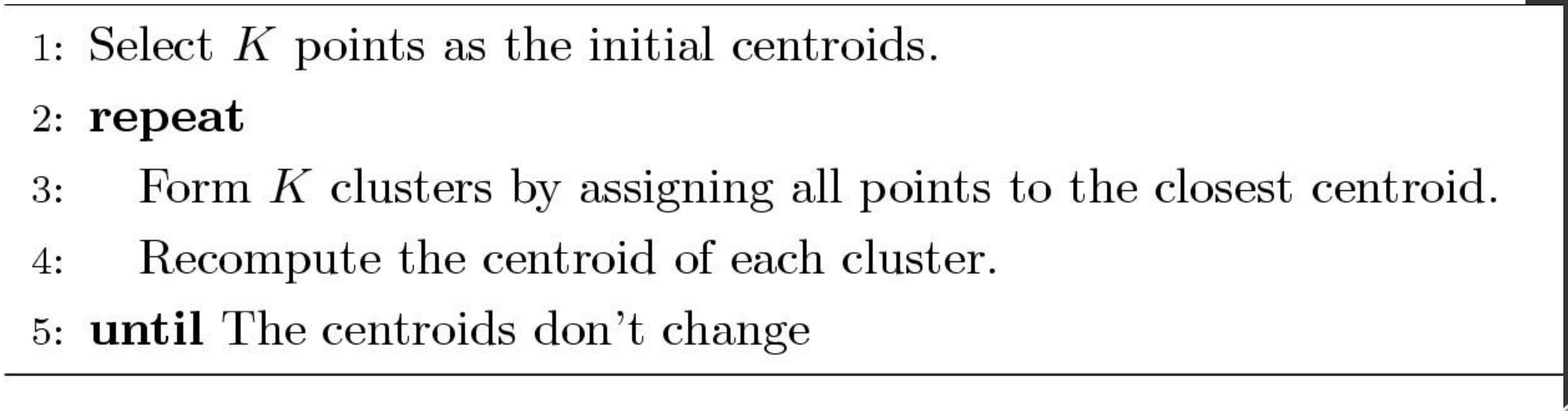
K-means clustering

- Given a *K*, find a partition of *K clusters* to optimize the chosen partitioning criterion (cost function)
  - global optimum: exhaustively search all partitions

- The *K-means* algorithm: a heuristic method
  - K-means algorithm (MacQueen'67): each cluster is represented by the centre of the cluster and the algorithm converges to stable centriods of clusters.
  - K-means algorithm is the simplest partitioning method for clustering analysis and widely used in data mining applications.

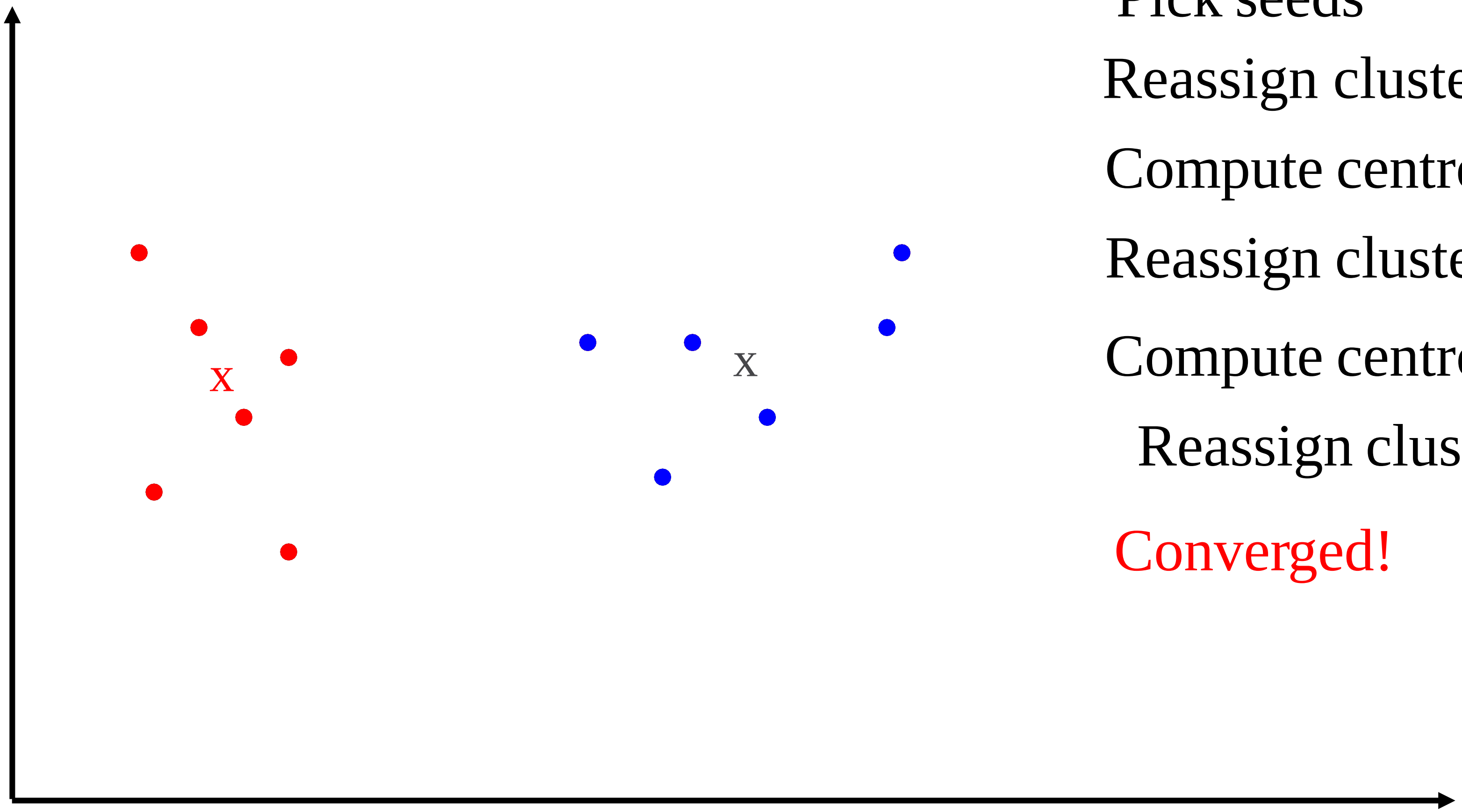
K-means Clustering

- Partition clustering approach
- Each cluster is associated with a **centroid (center point)**
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple



```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K Means Example (*K*=2)



Pick seeds

Reassign clusters

Compute centroids

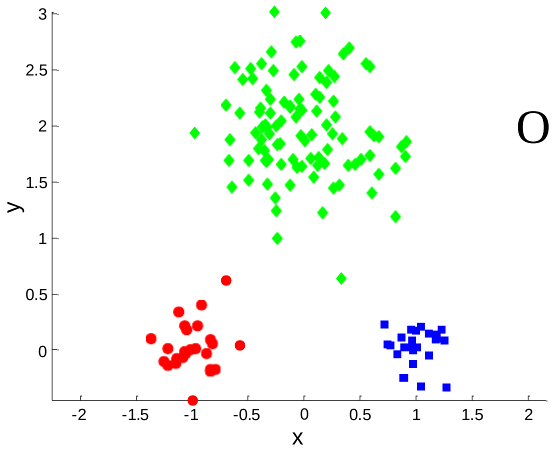
Reassign clusters

Compute centroids

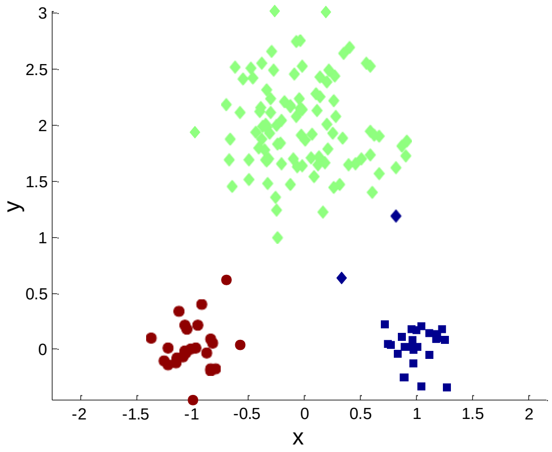
Reassign clusters

Converged!

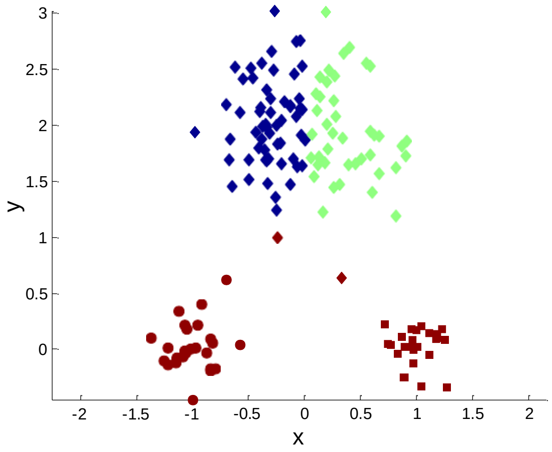
Two different K-means Clusterings



Original Points

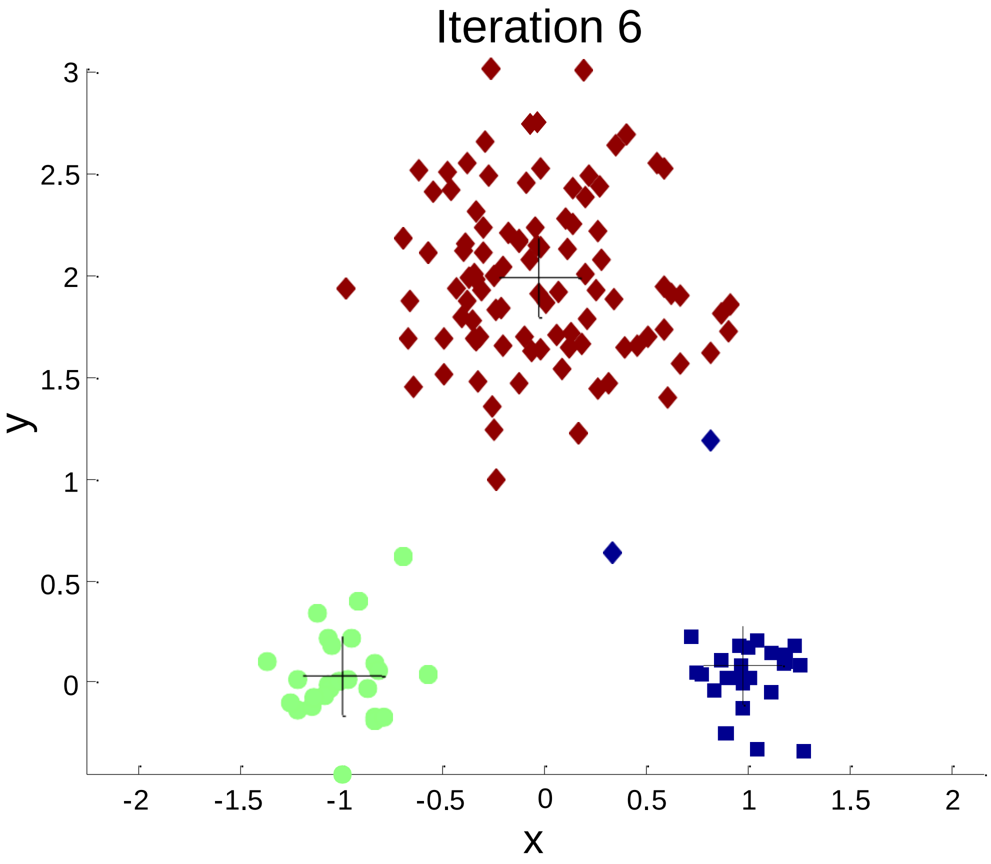


Optimal Clustering

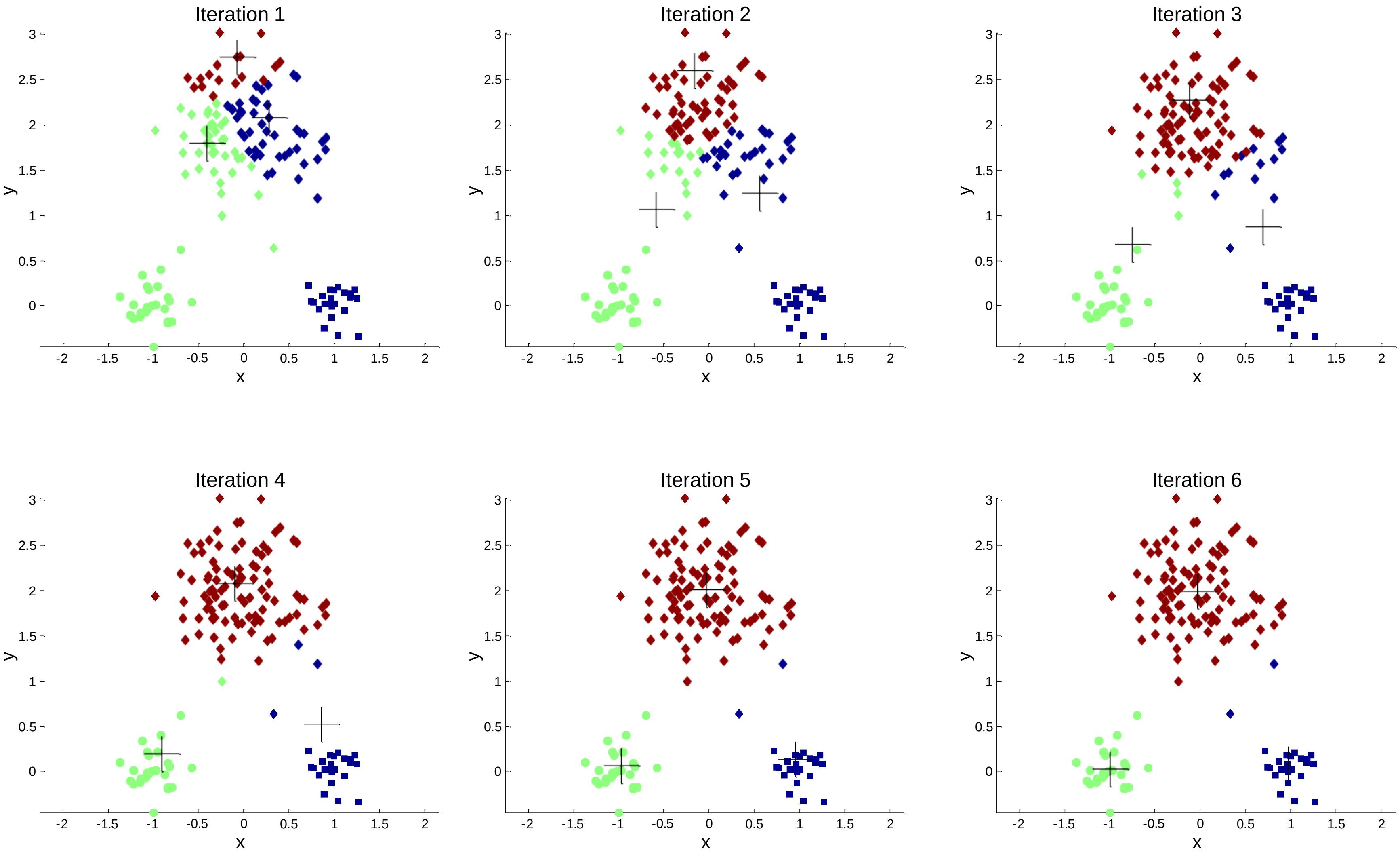


Sub-optimal Clustering

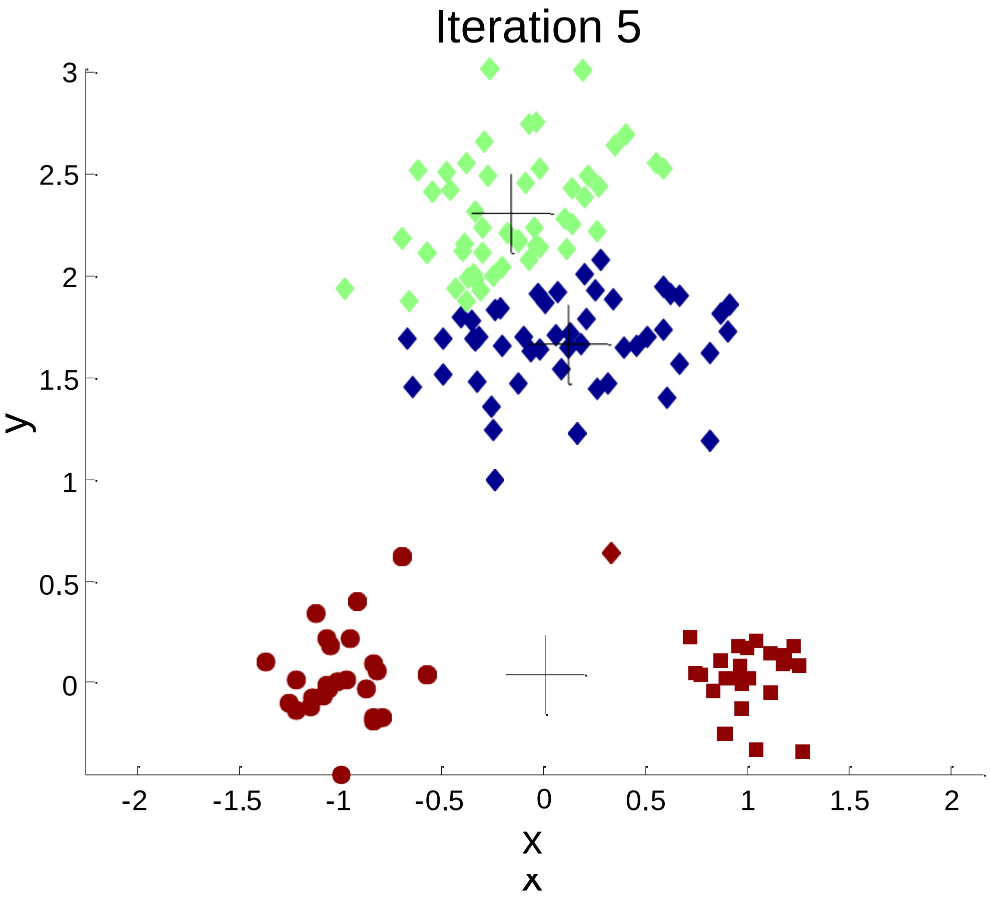
Importance of Choosing Initial Centroids



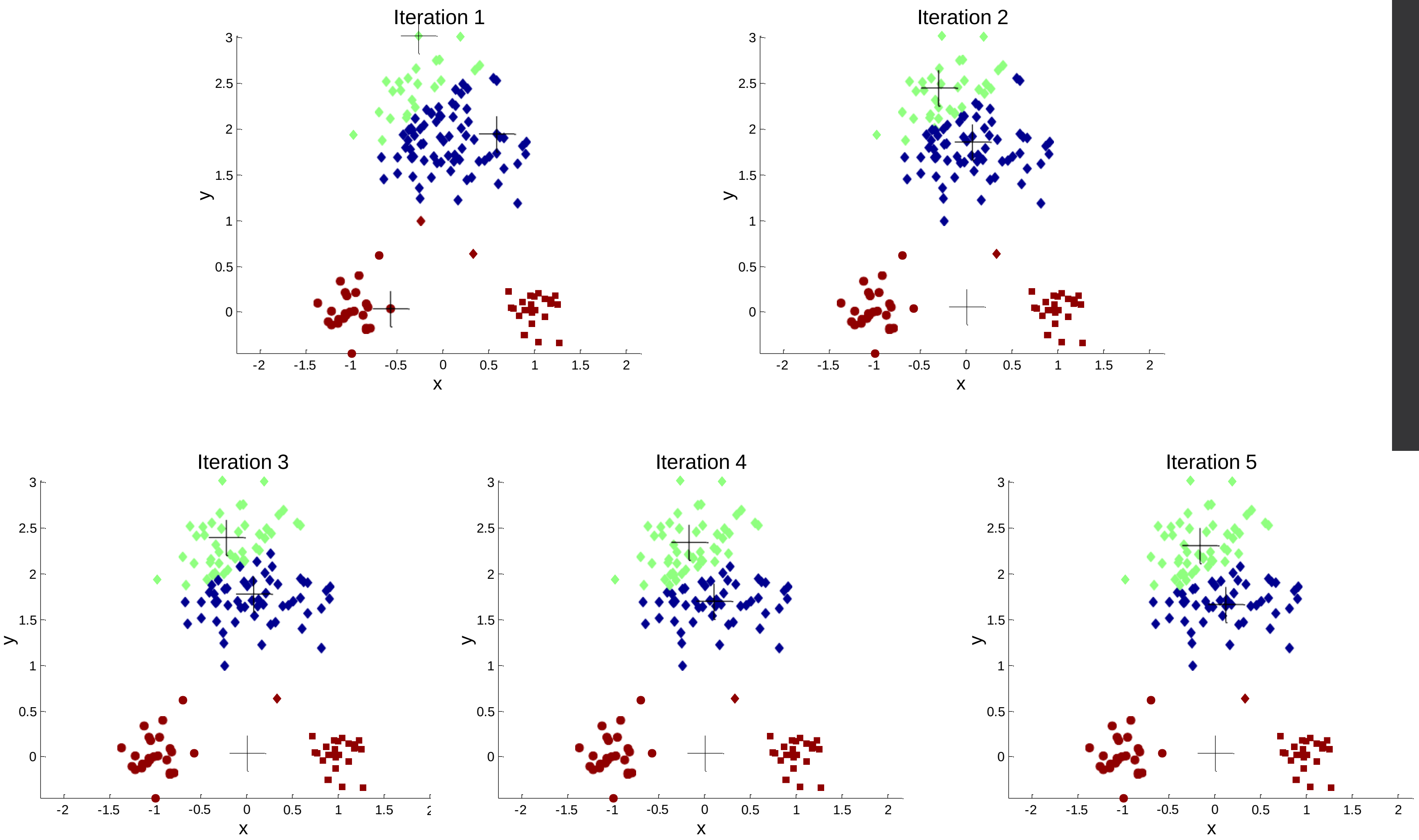
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



Other Limitations of K-means

- Dependence on Starting points for centroids
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.
- How to choose K?

K-Means Clustering in Python

- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- Required reading:
- Pg 168-181 Introduction to Machine Learning in Python