Discussion 13: PCA, Clustering

Data 100, Spring 2020

Suraj Rampure

Friday, May 1st, 2020

Agenda

- Taxonomy of ML
- PCA
- Clustering

As per usual, everything will be posted at

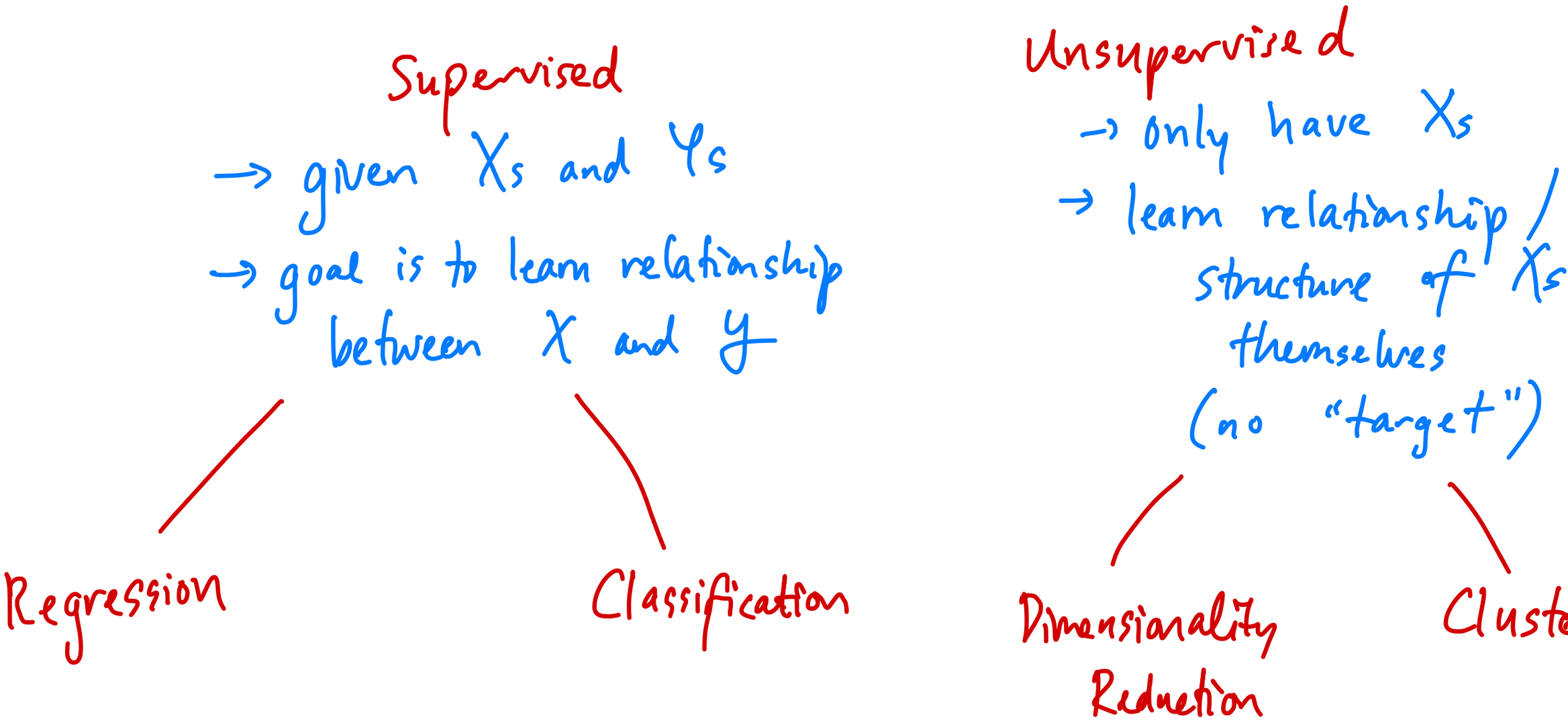
http://surajrampure.com/teaching/ds100.html

In addition, here's a feedback form:

http://tinyurl.com/feedbacksuraj

Taxonomy of Machine Learning

Before talking about PCA and clustering, let's talk about the two different branches of machine learning – **supervised** and **unsupervised** learning.



Dimensionality Reduction

Why may we want to reduce the dimensionality of our data?

→ hard to visualize many features at once

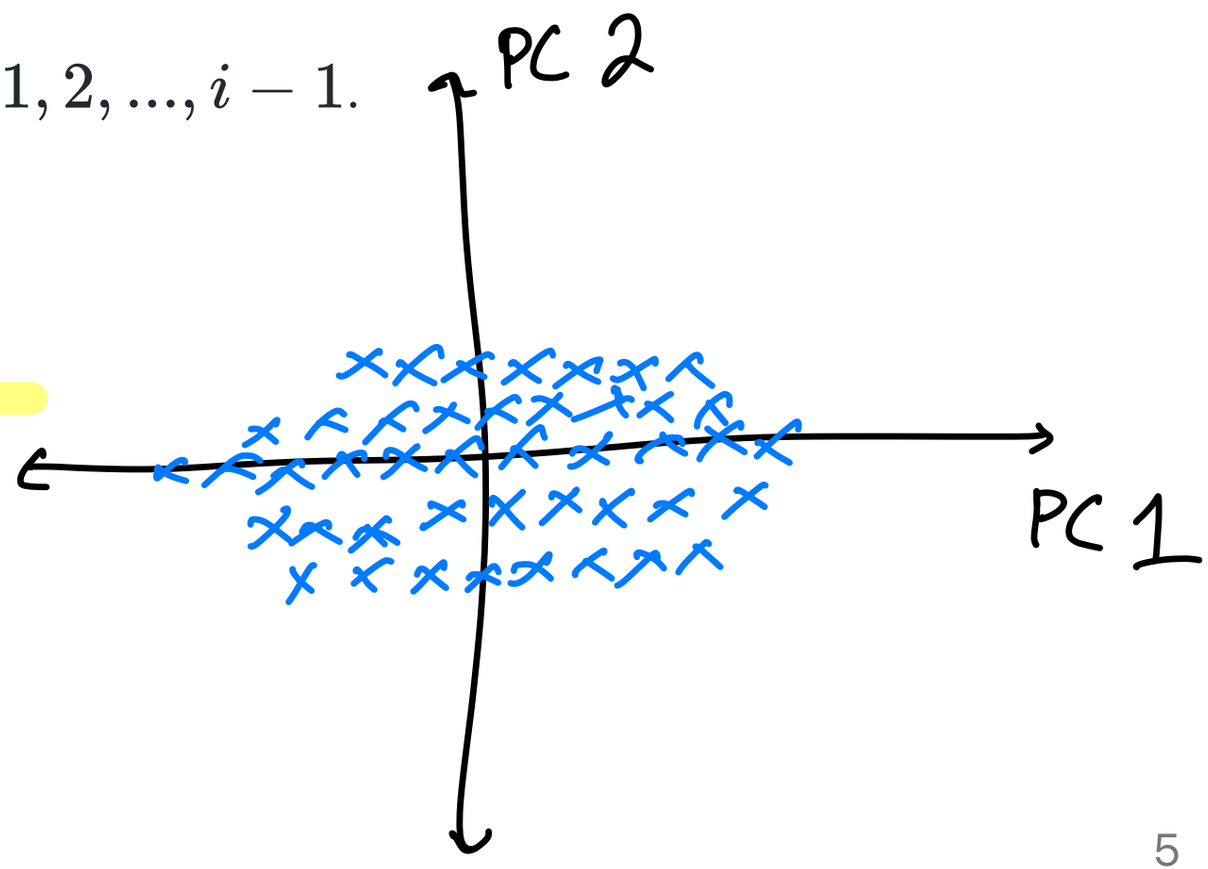
→ many features may not be useful

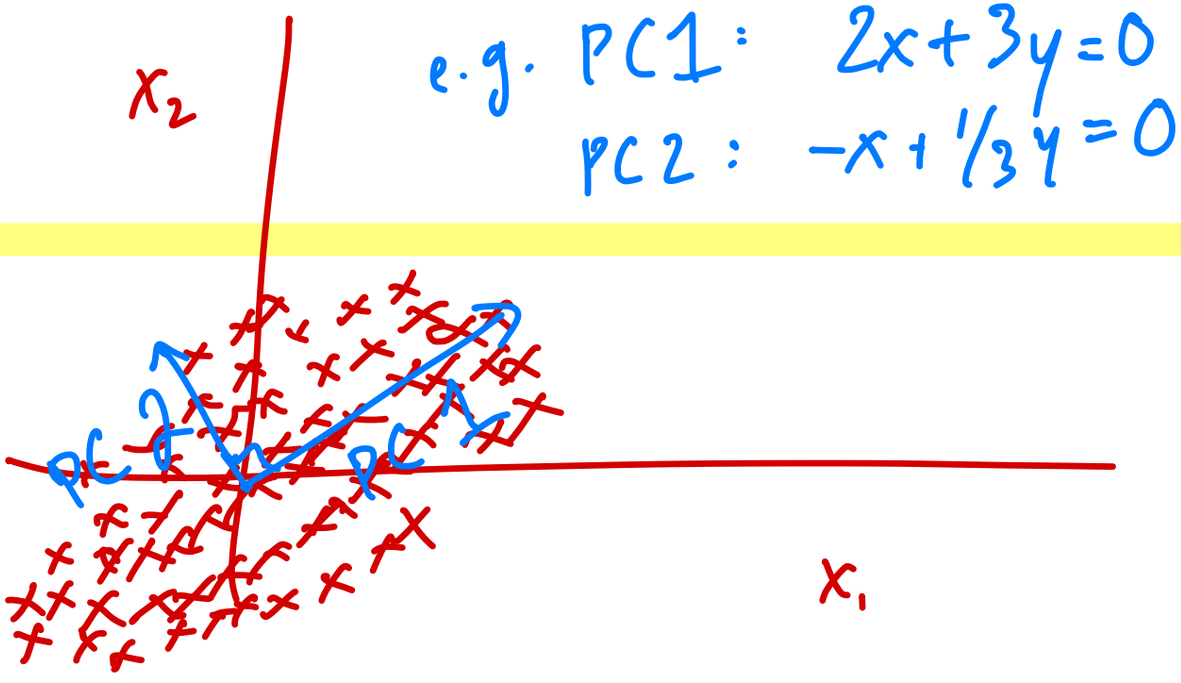
Simplest method of dim red:
- pick some subset of features
- but that throws away a lot of info!

PCA

PCA is a dimensionality reduction technique. Its goal is to find a **set of orthogonal basis vectors ("axes", or "directions") within the data that maximize variance**. Projections onto these new axes are called **principal components**.

- The first "axis" should capture the most variance possible.
- The second "axis" should capture the most variance possible, **given that it is orthogonal to the first axis**.
- In general, axis i should be orthogonal to axes $1, 2, ..., i-1$.



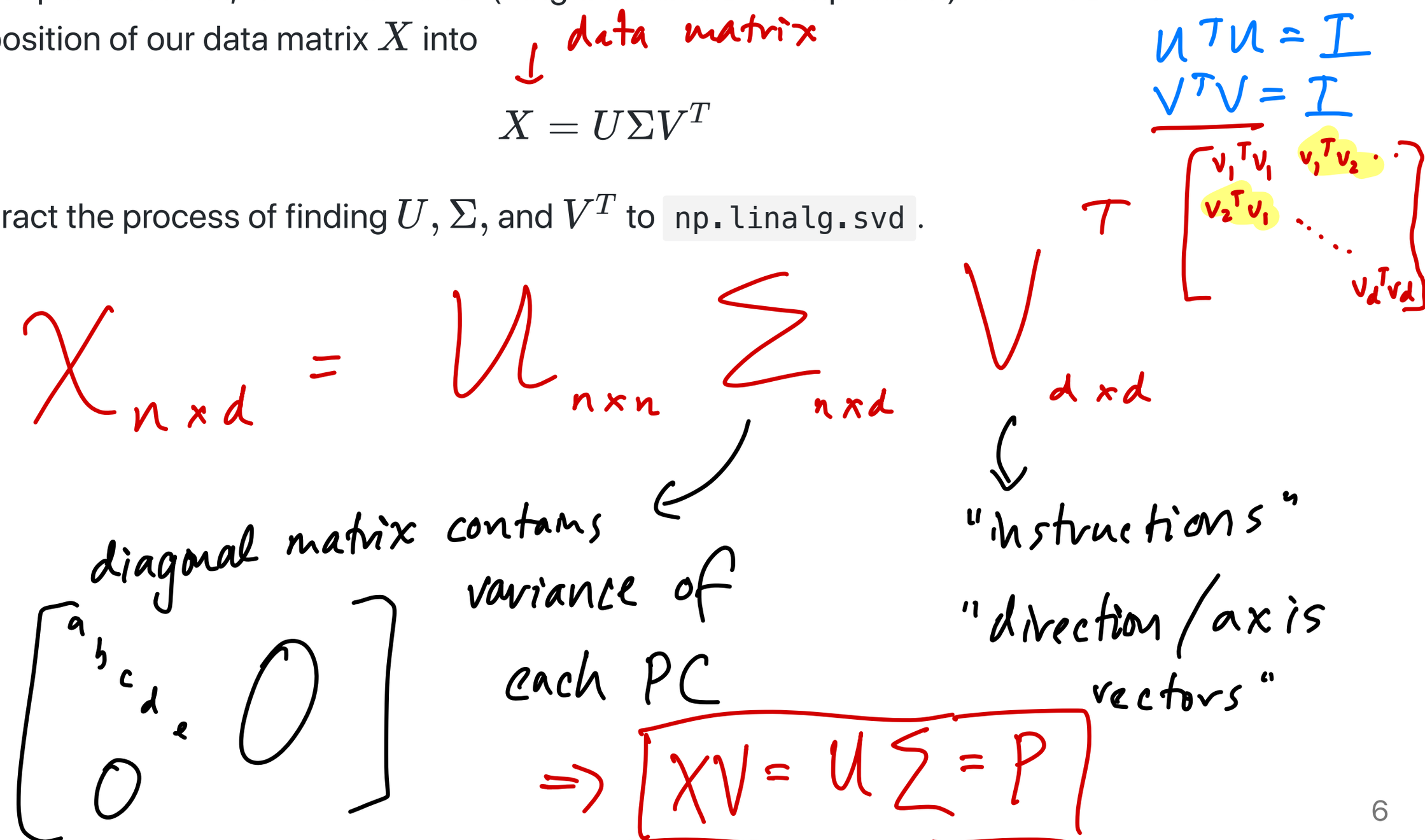


PCA and the Singular Value Decomposition

In order to perform PCA, we use the SVD (Singular Value Decomposition). The SVD is the decomposition of our data matrix X into

$$X = U\Sigma V^T$$

We abstract the process of finding U, Σ, and V^T to `np.linalg.svd`.



$$P = X V = U \Sigma$$

$$P = \begin{bmatrix} | & | & \\ PC1 & PC2 & \cdots \\ | & | & \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} s_1 & & \bigcirc \\ & s_2 & \\ \bigcirc & & \ddots \\ & & & s_p \end{bmatrix}$$

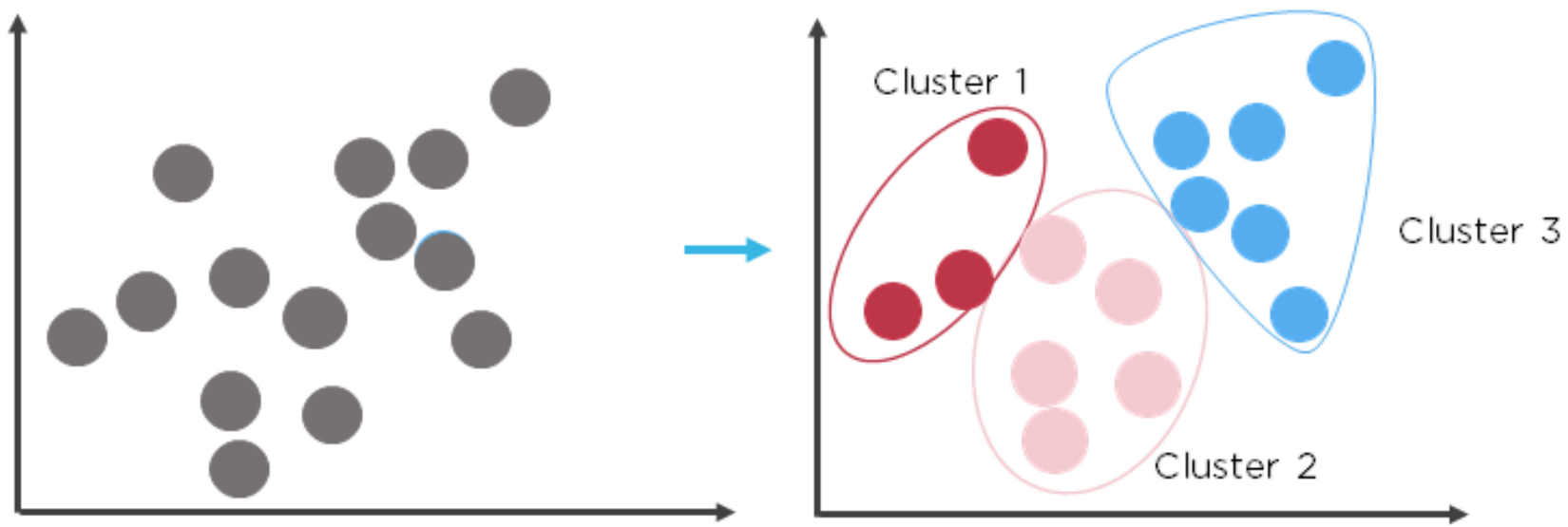
```
plt.scatter(P[:, 0],
            P[:, 1])
```

$$Var(PC\ i) = \frac{s_i^2}{N}$$

Clustering

Clustering is an **unsupervised** learning task. We're given raw data, and we want to try and find clusters in it. This can be useful in performing EDA (understanding the data we're given), or even in perhaps performing classification.

In clustering, we want to assign our data points to one of some number of clusters.



K-Means Clustering

In k-means clustering, we **first** pick k, the number of clusters that we're trying to find.

- The principle that k-means clustering follows is that points in a given cluster should be close to the **centroid** of that cluster.

Algorithm:

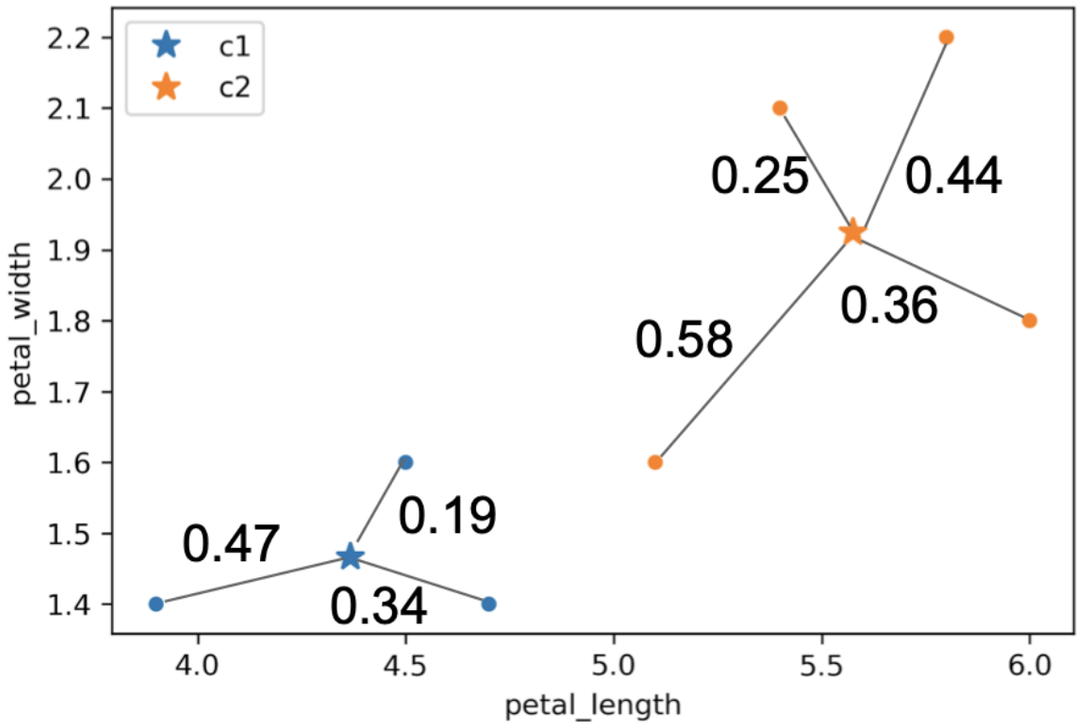
```
initialize k centroids randomly

repeat until convergence (no change in clusters):
        assign each point to the closest centroid
        update centroids based on new clusters
```

Nice visualization: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Distortion

Distortion is one of the possible loss functions we can use for clustering, and it is the one that k-means clustering minimizes.



Distortion is the **sum of the average of the squared distance to the centroid for each cluster**:

$$D = \frac{0.47^2 + 0.19^2 + 0.34^2}{3} + \frac{0.58^2 + 0.25^2 + 0.44^2 + 0.36^2}{4}$$

Agglomerative Clustering

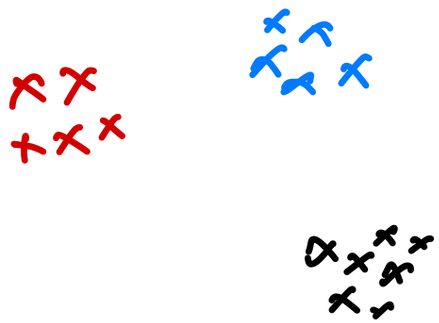
Agglomerative clustering is another clustering technique. Similarly to k-means, we also start off with a fixed number k of clusters that we want to find.

Algorithm:

```
assign each point to its own cluster

while there are more than k clusters left:
        join the two closest clusters
```

Question: What does it mean for two clusters to be the "closest" together?



Silhouette Score

The silhouette score is a metric for how good the cluster assignment **for a single point** is (whereas distortion was for an entire dataset).

Given the following definitions,

$$A = \text{average distance to points in its own cluster}$$

$$B = \text{average distance to points in the next nearest cluster}$$

We define S as

$$S = \frac{B - A}{\max(A, B)}$$

$-1 \leq S \leq 1$