



# Lecture 14: Beyond K Means Clustering

# Recap

- Kmeans algorithm
- Expectation Maximization
- Centroid initialization with kmeans++
- Kmeans decision boundary
- Kmeans limitations with oblong data clusters
- Kmeans limitations with outliers
- Using Kmeans and silhouette analysis for outlier detection



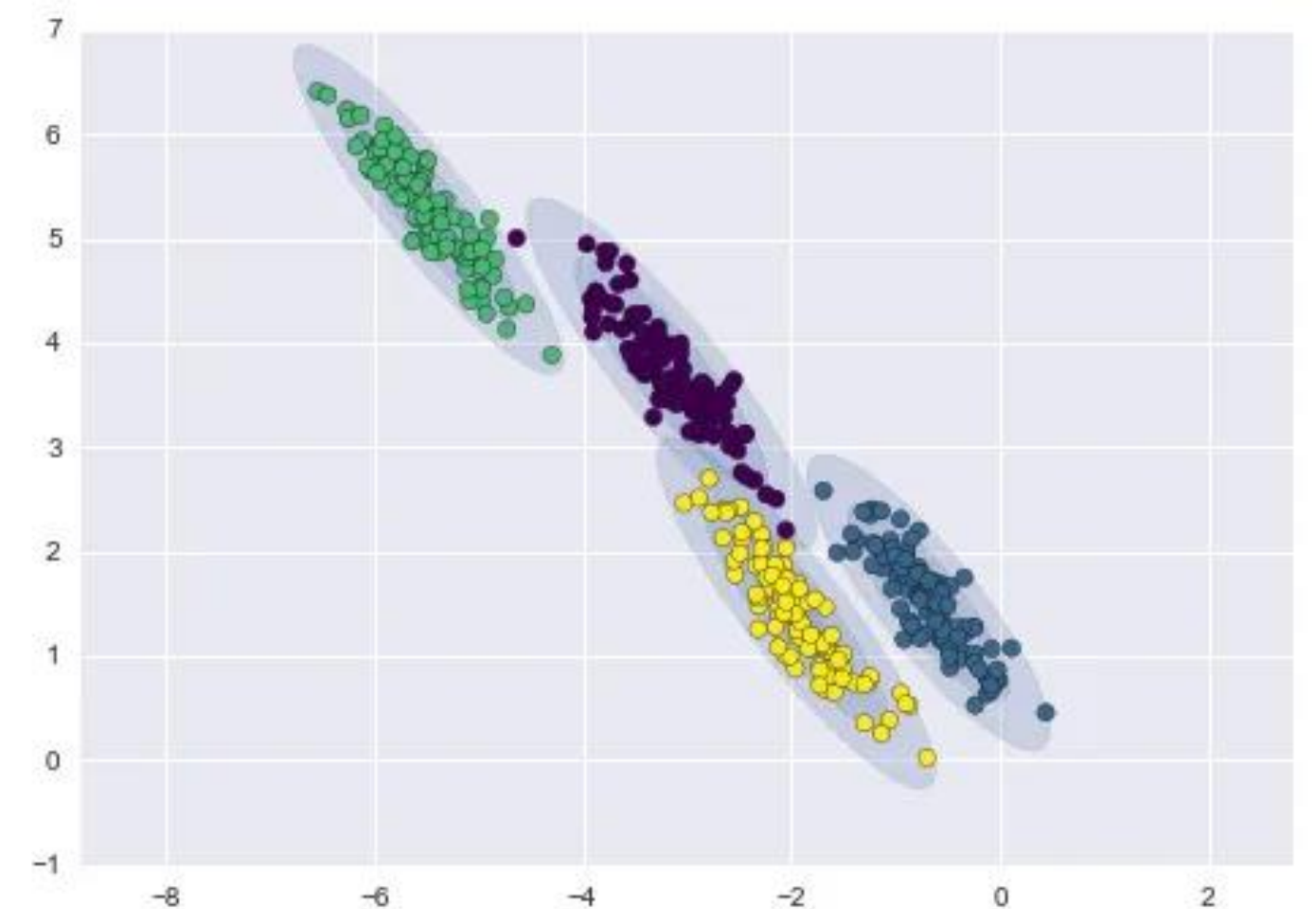


# Another way of looking at clusters

- Hard versus soft clustering
- Hard:
  - Clusters don't overlap
  - Elements belong to a cluster or they don't
  - E.g. K-Means
- Soft
  - Clusters may overlap
  - Strength of association between cluster and instances
  - 60% confidence of cluster1 membership, 40% cluster2
  - E.g. Gaussian Mixture Model Clustering

# Mixture Model clustering

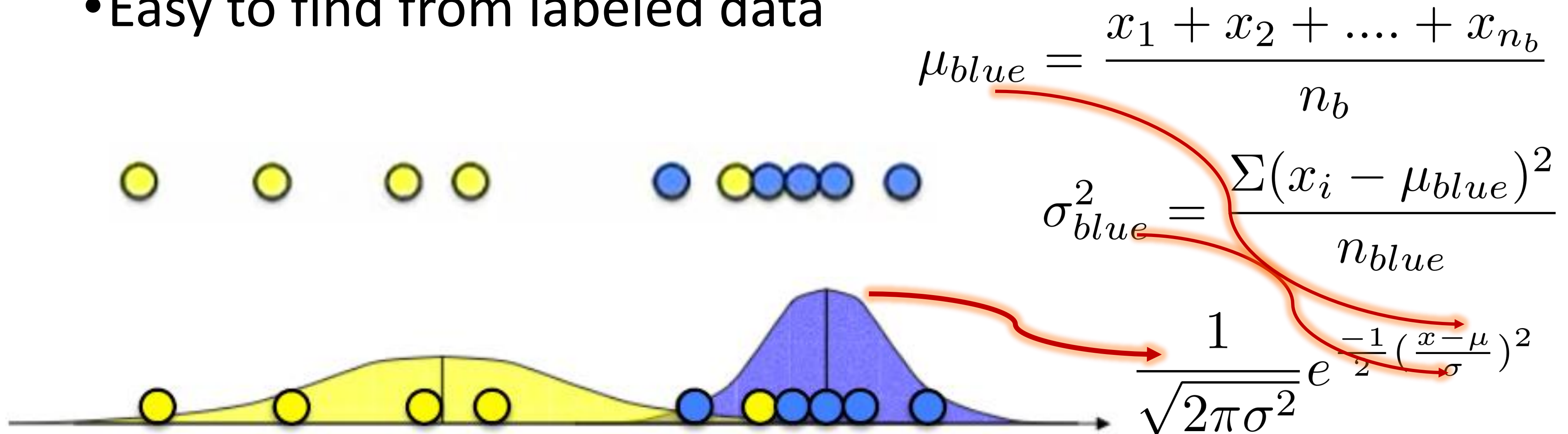
- Probabilistic way of doing soft clustering
  - Soft clustering - Cluster membership is not 0/1
  - Probability
- Each cluster is a generative model (probability distribution)
- But parameters (mean and covariance) are unknown
  - Discovered as part of clustering
- Parameters are discovered by EM
  - EM in GMM is EM in true sense 😊





# Recap: Estimate Mean & variance in generative ML

- We are given data  $x_1 \dots x_n$ 
  - Mean and variance not known
  - Easy to find from labeled data



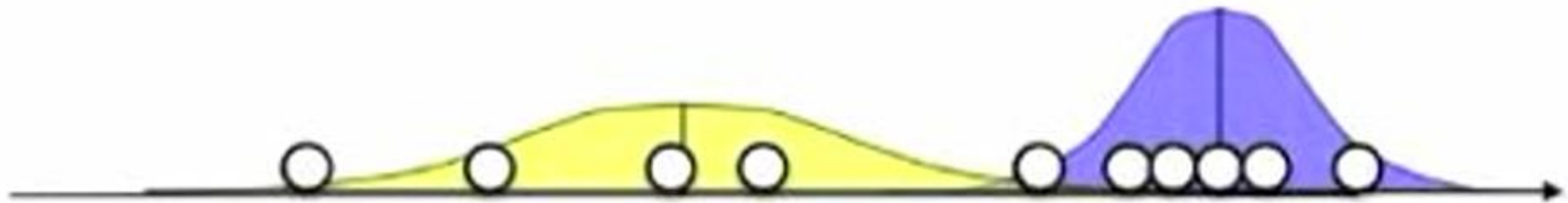
- What if we didn't know which data came from which distribution? (unsupervised learning)

# Gaussian Mixture Models

- Chicken & Egg
  - We need labels to determine mean, variance



- We need mean, variance to predict labels



- If we knew mean/var, we could guess which point came from which Gaussian. But we don't know

# Gaussian Mixture Models Intuition

- We are given data points
- They came from  $k$  Gaussian distributions
- We don't know which point came from which Gaussian



- Solution: EM
- Start with two Gaussian with random mean and variances
  - Just like kmeans centroids





# Gaussian Mixture Models EM

- 2 Gaussian with random mean and variances
- For each point  $x_i$ , does this look like it came from a or b?



- Soft assignment
- No max P calc

$$P(x^{(i)}|b) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \left( \frac{x^{(i)} - \mu_b}{\sigma_b} \right)^2}$$

$$P(b|x^{(i)}) = \frac{P(x^{(i)}|b)P(b)}{P(x^{(i)}|b)P(b) + P(x^{(i)}|a)P(a)}$$

$$P(a|x^{(i)}) = \frac{P(x^{(i)}|a)P(a)}{P(x^{(i)}|b)P(b) + P(x^{(i)}|a)P(a)}$$

# Gaussian Mixture Models EM



- Rinse and Repeat

$$p_{b_i} = P(b|x^{(i)}) = \frac{P(x^{(i)}|b)P(b)}{P(x^{(i)})}$$

$$p_{a_i} = P(a|x^{(i)}) = \frac{P(x^{(i)}|a)P(a)}{P(x^{(i)})}$$

- Calculate the new mean/var

$$\mu_b = \frac{p_{b_1}x^{(1)} + p_{b_2}x^{(2)} + \dots + p_{b_n}x^{(n)}}{p_{b_1} + p_{b_2} + \dots + p_{b_n}}$$

$$\mu_a = \frac{p_{a_1}x^{(1)} + p_{a_2}x^{(2)} + \dots + p_{a_n}x^{(n)}}{p_{a_1} + p_{a_2} + \dots + p_{a_n}}$$

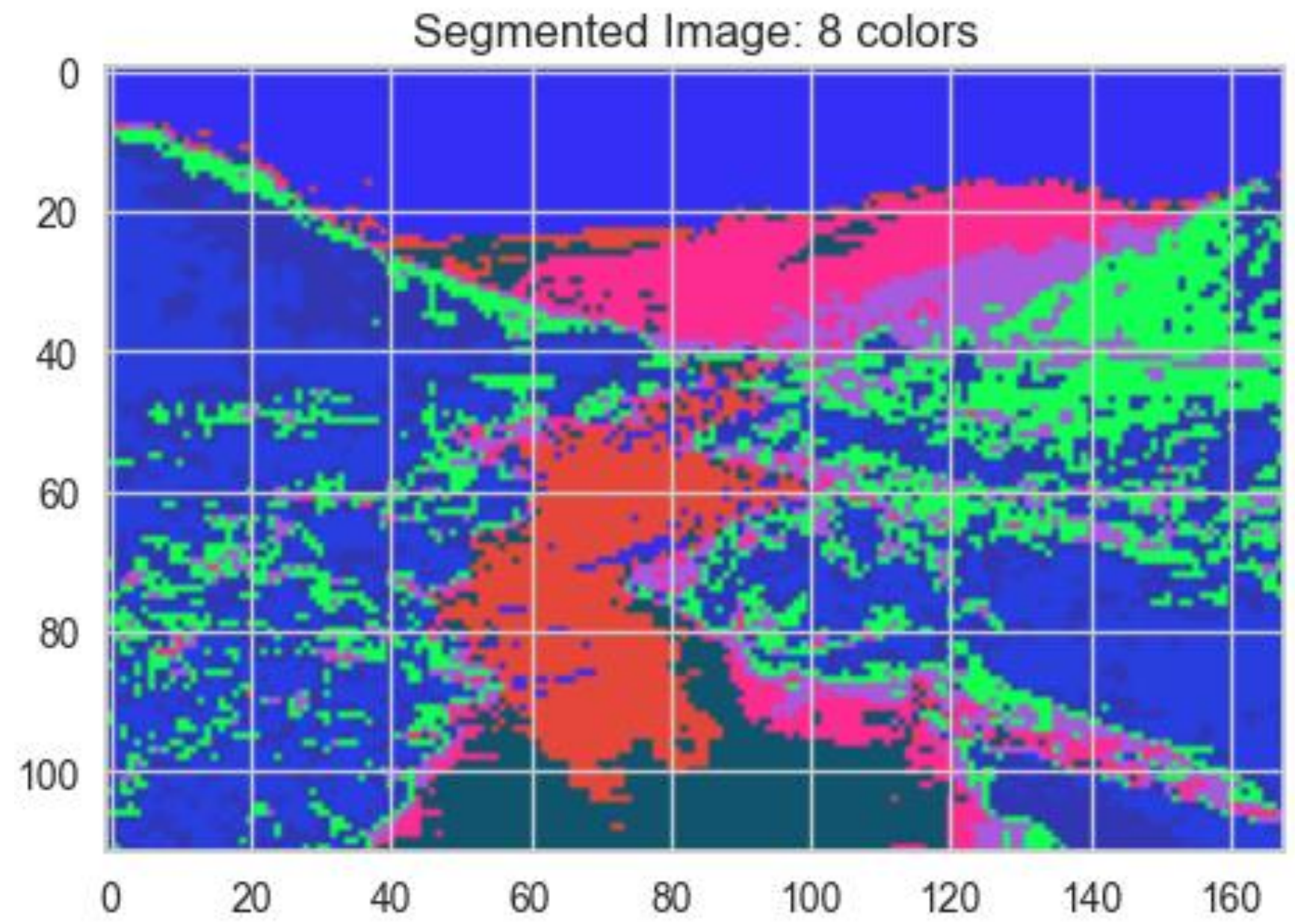
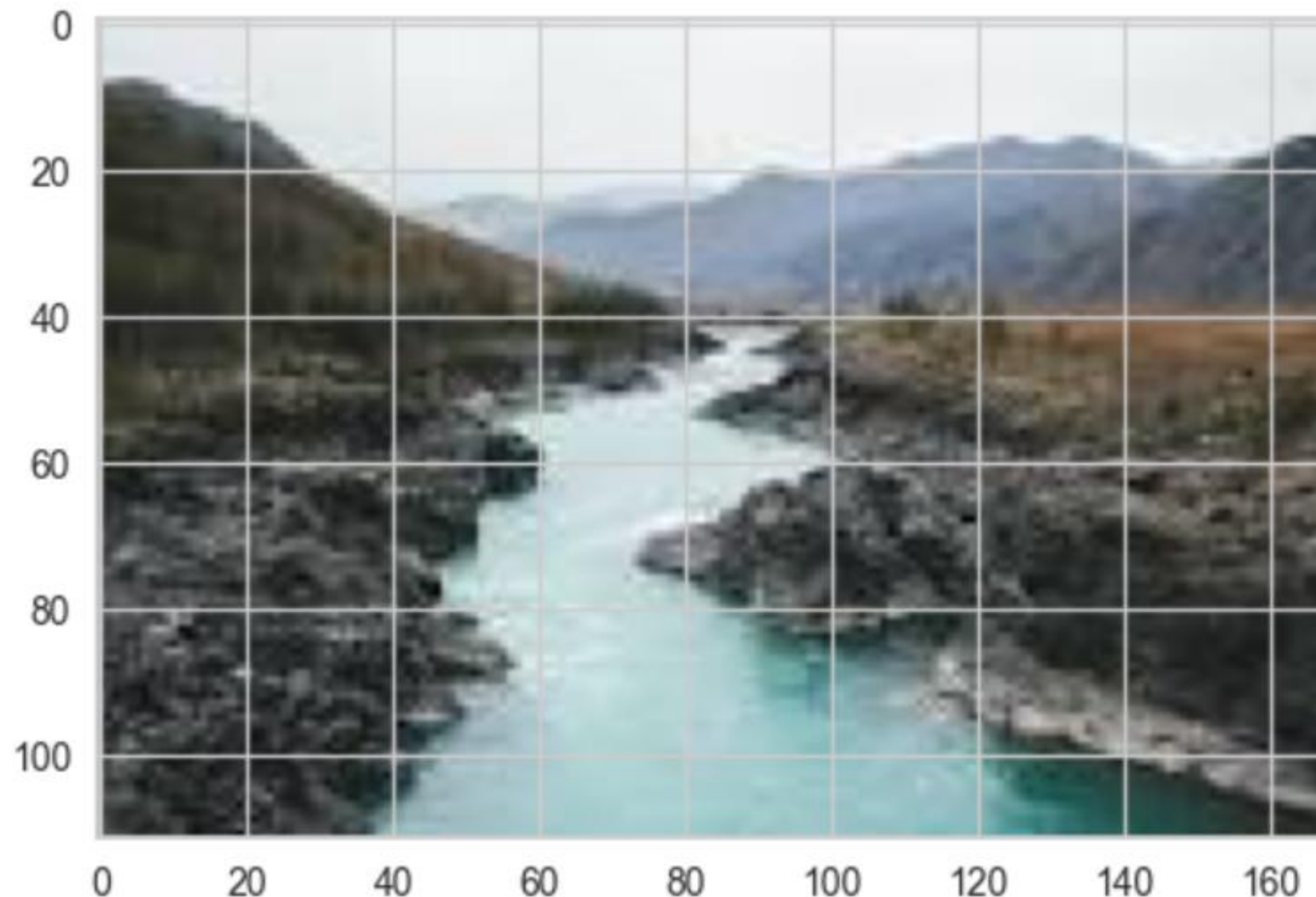
# GMM clustering applications

- Real life scenarios are mixtures, never black and white
- Finance
  - Investment Portfolio construction
  - Identifying stocks for growth (outliers in a good way)



# GMM clustering applications

- Image segmentation (less expensive way)
  - Split into patches & cluster based on image characteristics
  - Medical image – locate specific structures



# GMM clustering applications

- One tool of many towards explainable solution
  - Population to 2 wheeler, 4 wheeler ratio
  - Accident stats, Poisson mixtures for pattern identification



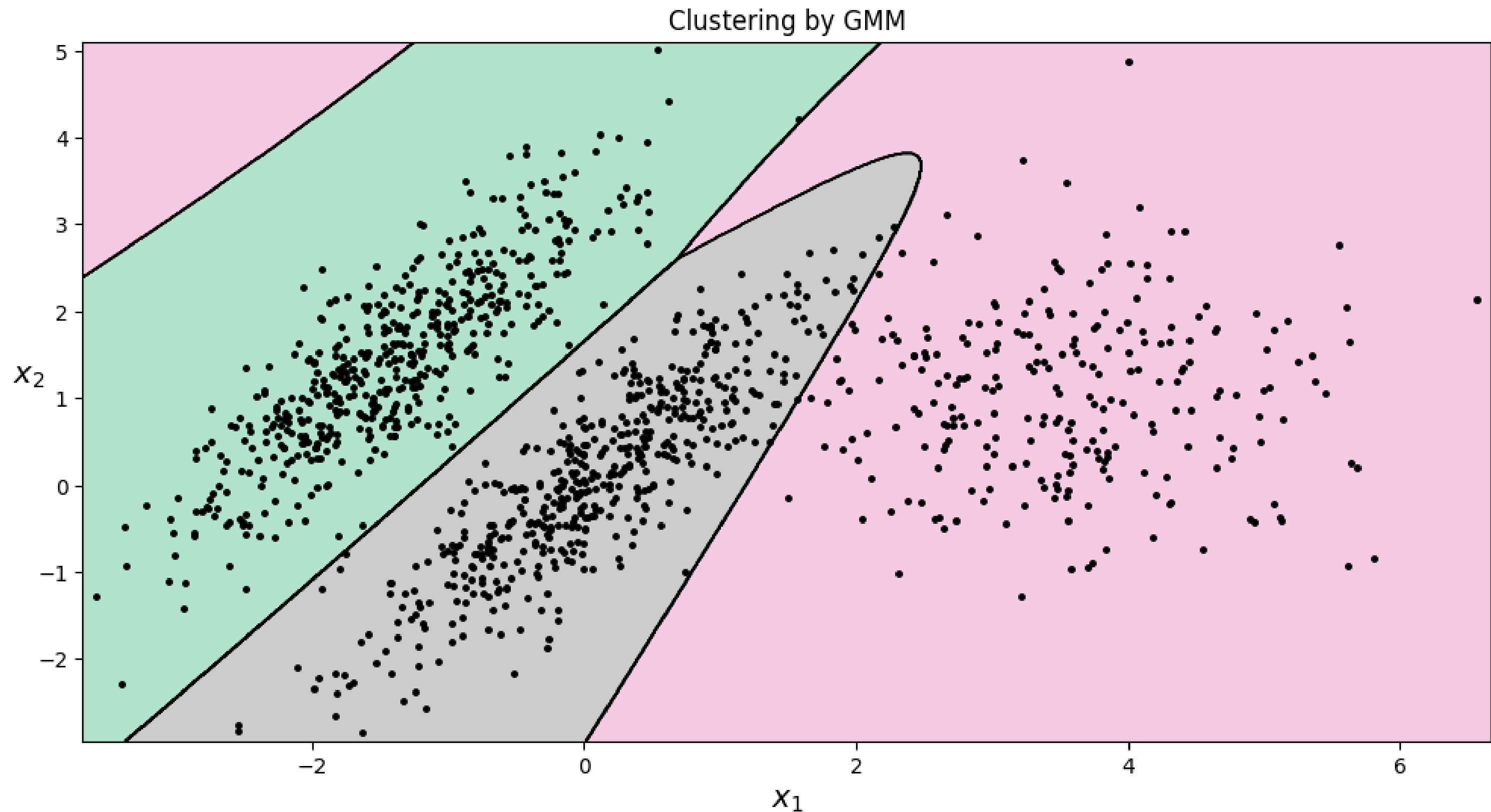
$$p_{b_i} = P(b|x^{(i)}) = \frac{P(x^{(i)}|b)P(b)}{P(x^{(i)})}$$

$$p_{a_i} = P(a|x^{(i)}) = \frac{P(x^{(i)}|a)P(a)}{P(x^{(i)})}$$

$$P(x^{(i)}|b) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \left( \frac{x^{(i)} - \mu_b}{\sigma_b} \right)^2}$$

# GMM decision boundary

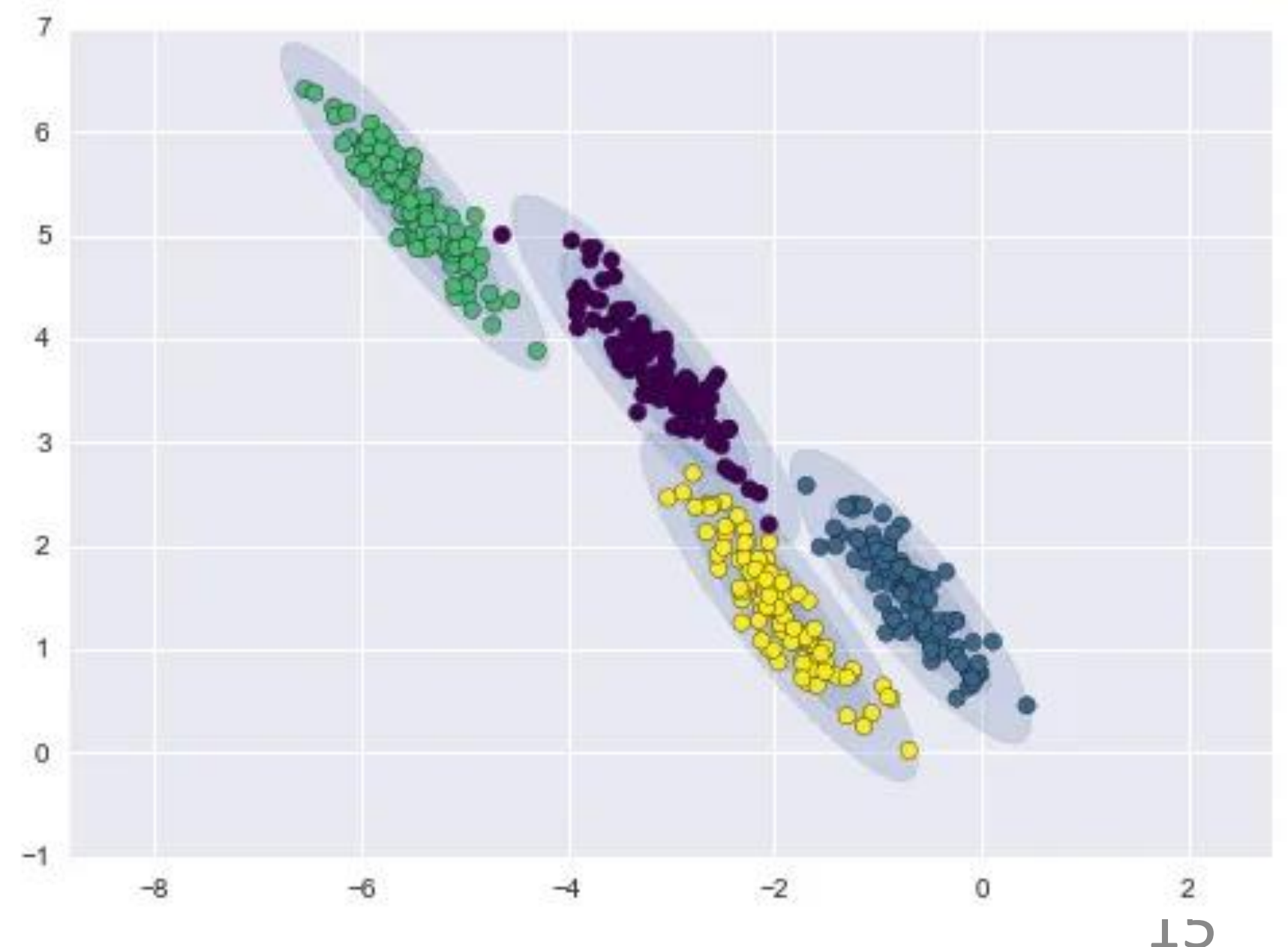
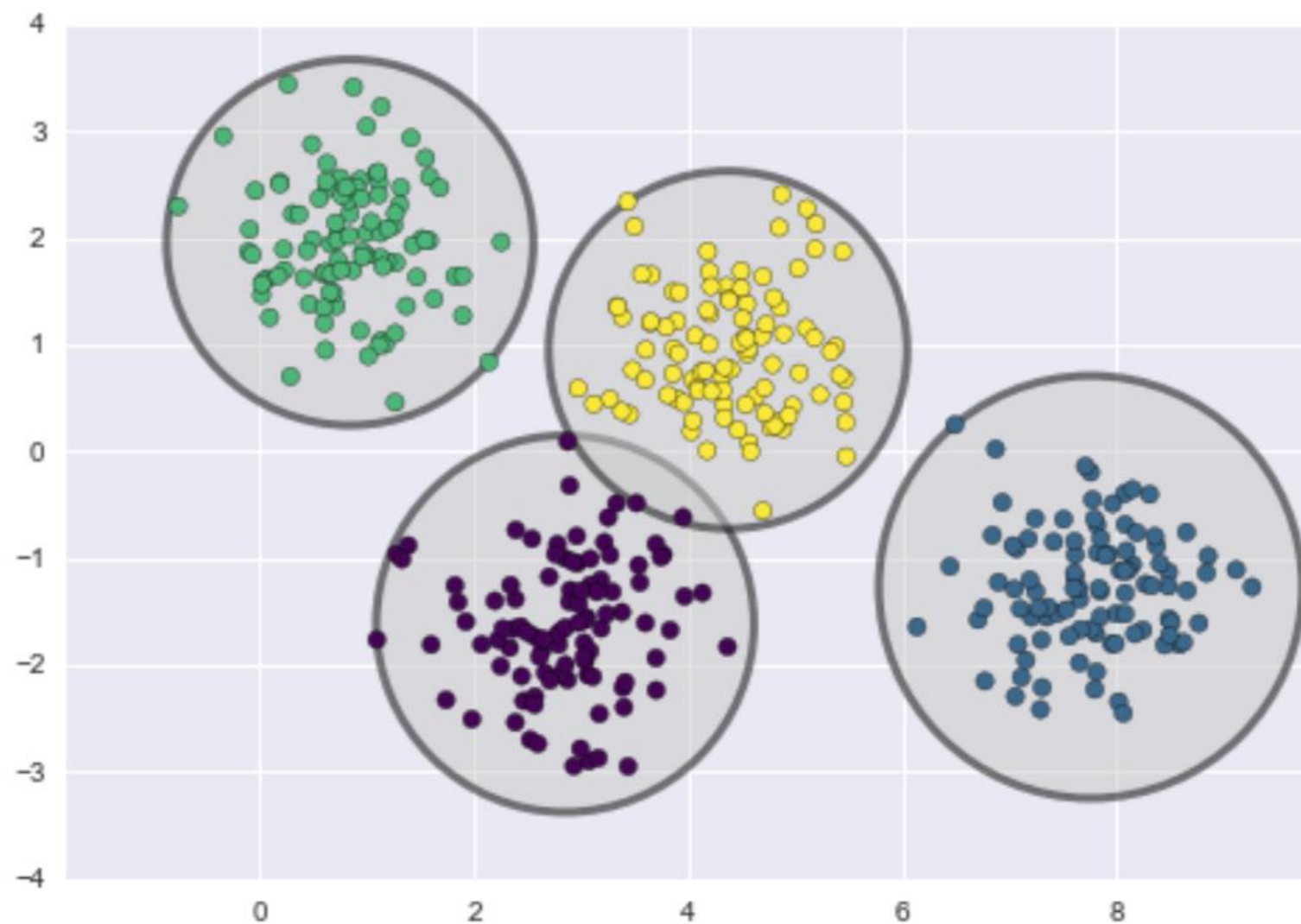
- Quadratic decision boundary





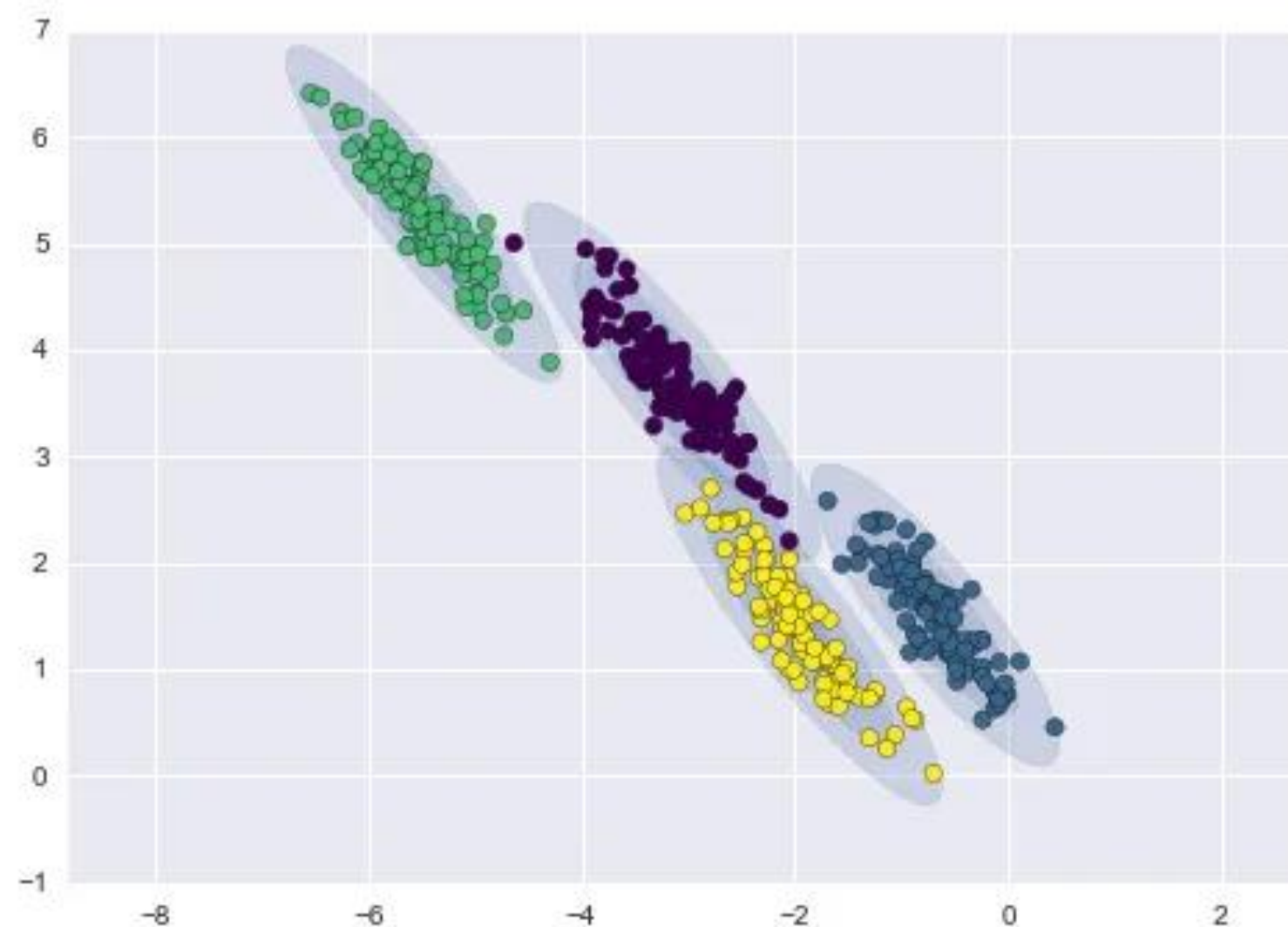
# GMM model

- Clustering every time some new data comes is expensive
- Could use extracted mean, variance, covariance as model
  - Multivariate Gaussian model from scratch!!



# Handling outliers with GMM

- GMM is sensitive to outliers
- Can use multivariate Gaussian techniques (Mahalanobis distance) or silhouette analysis
- Can use Minimum Covariance Determinant also







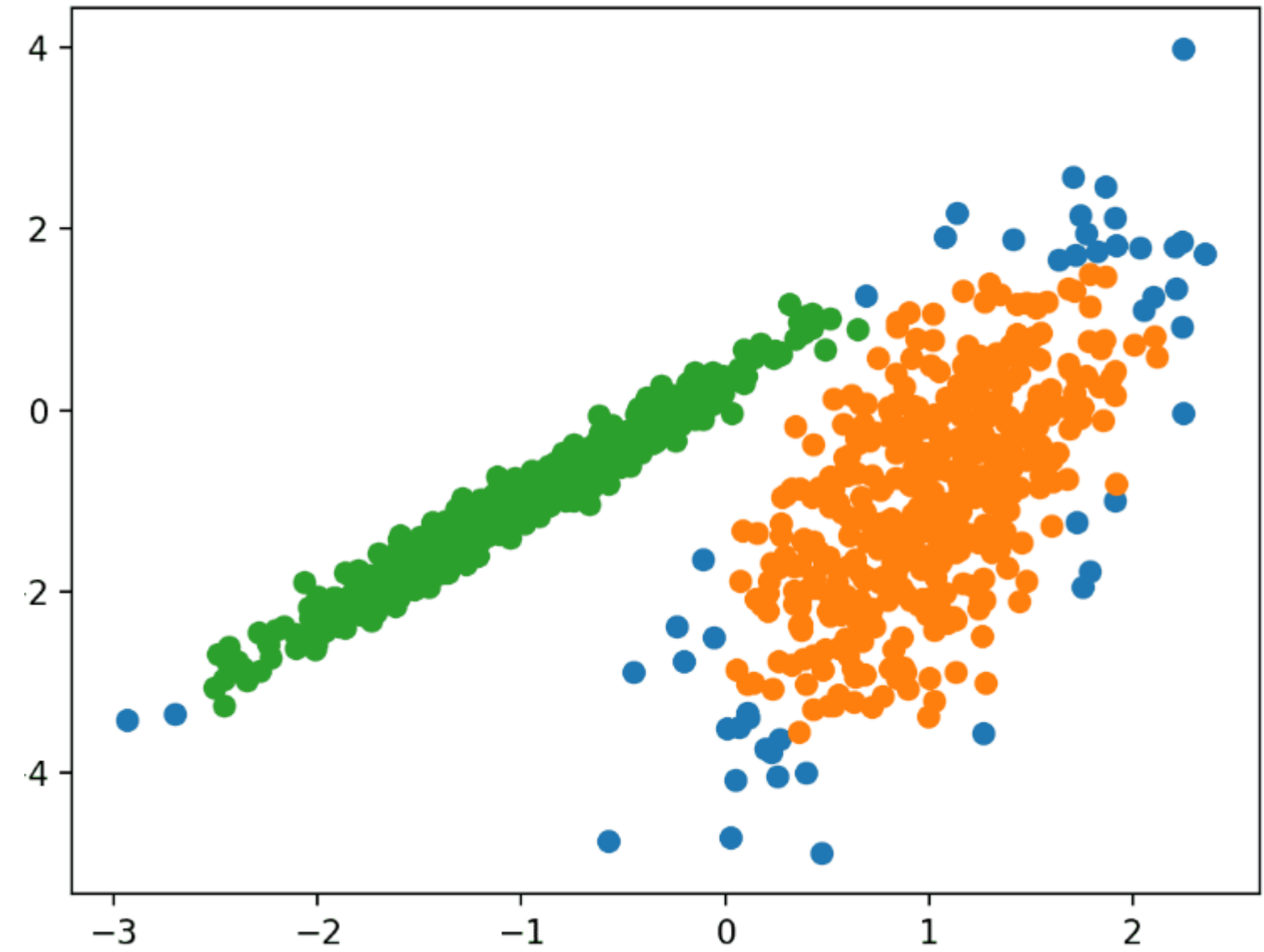
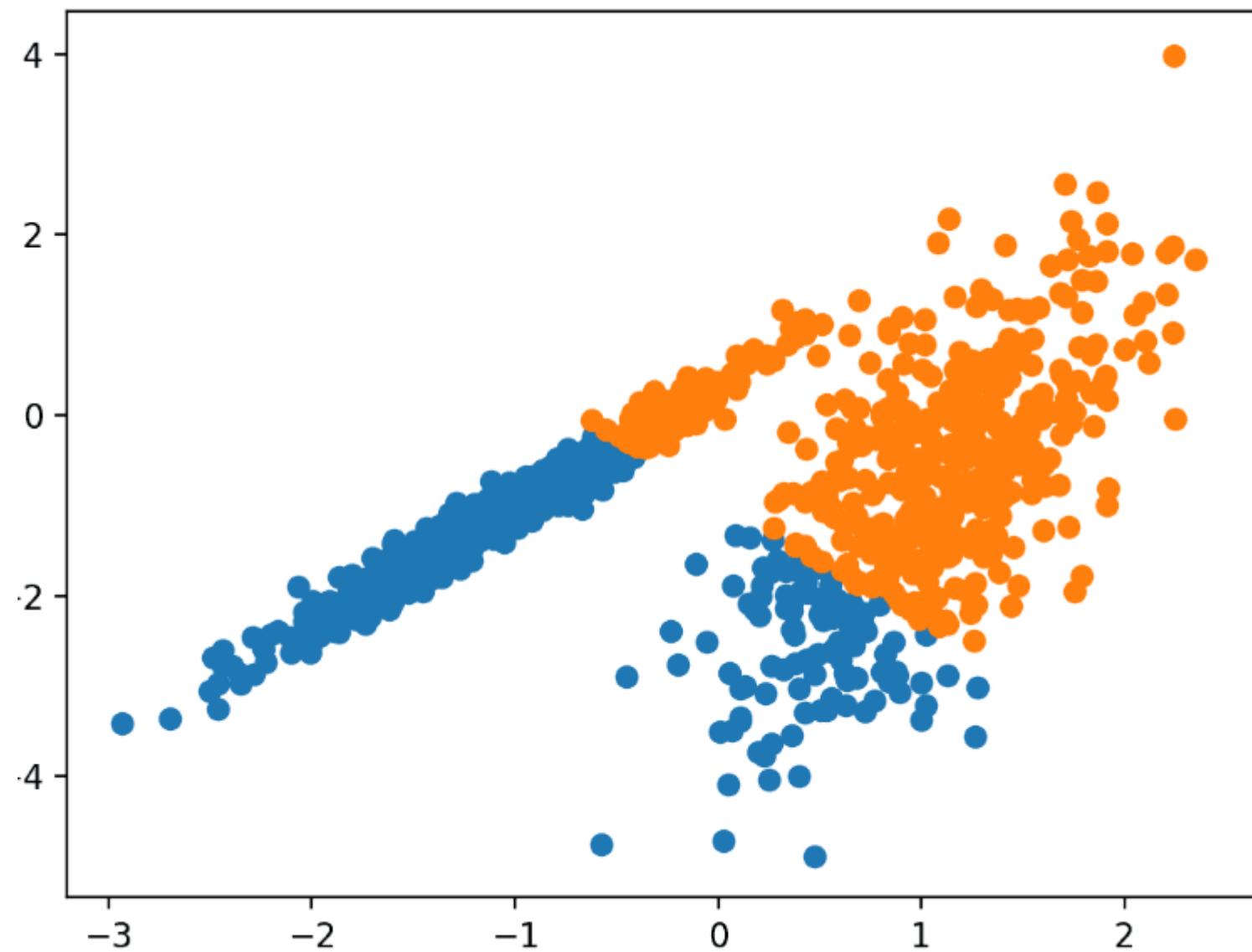
# Beyond K-Means & GMM



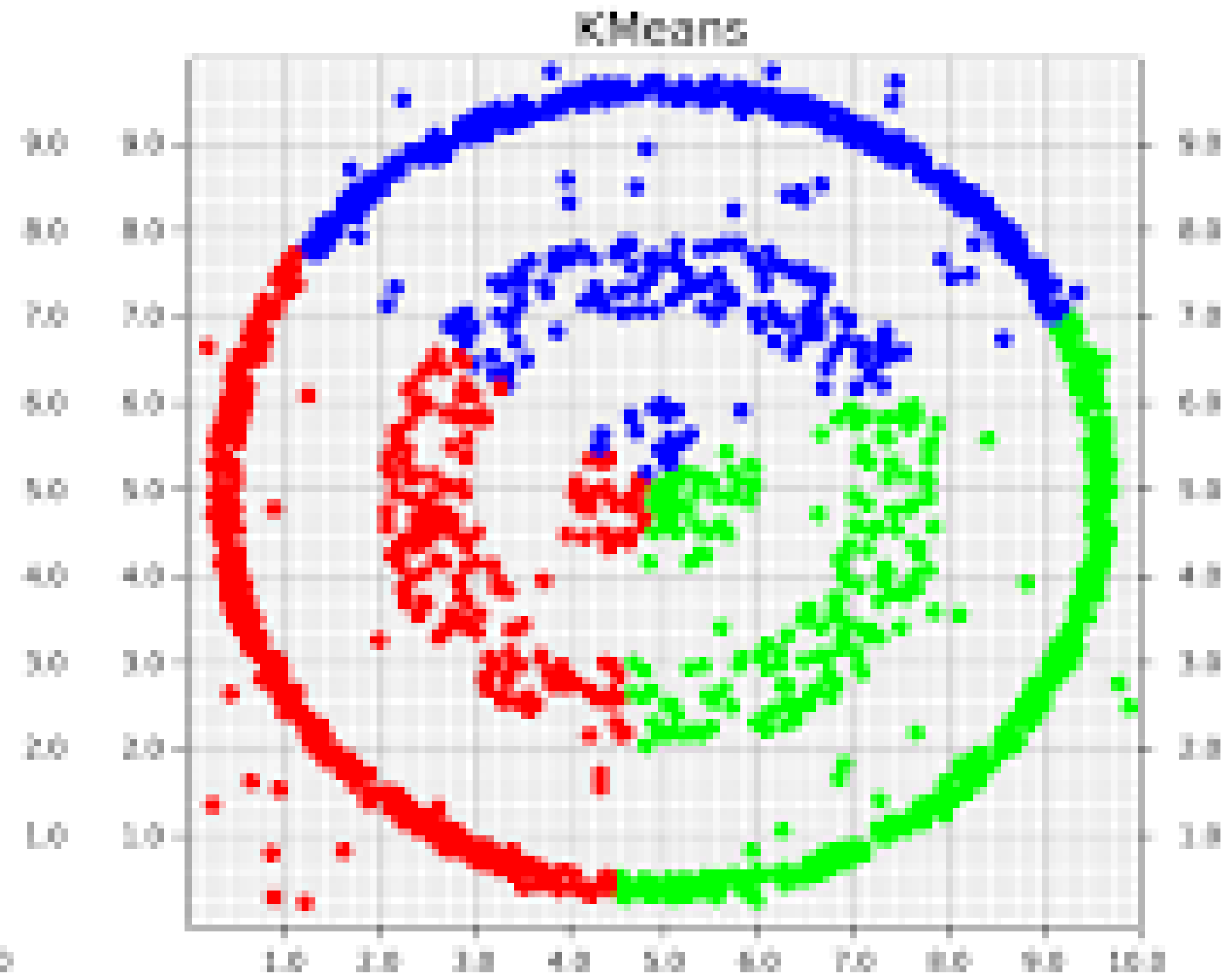
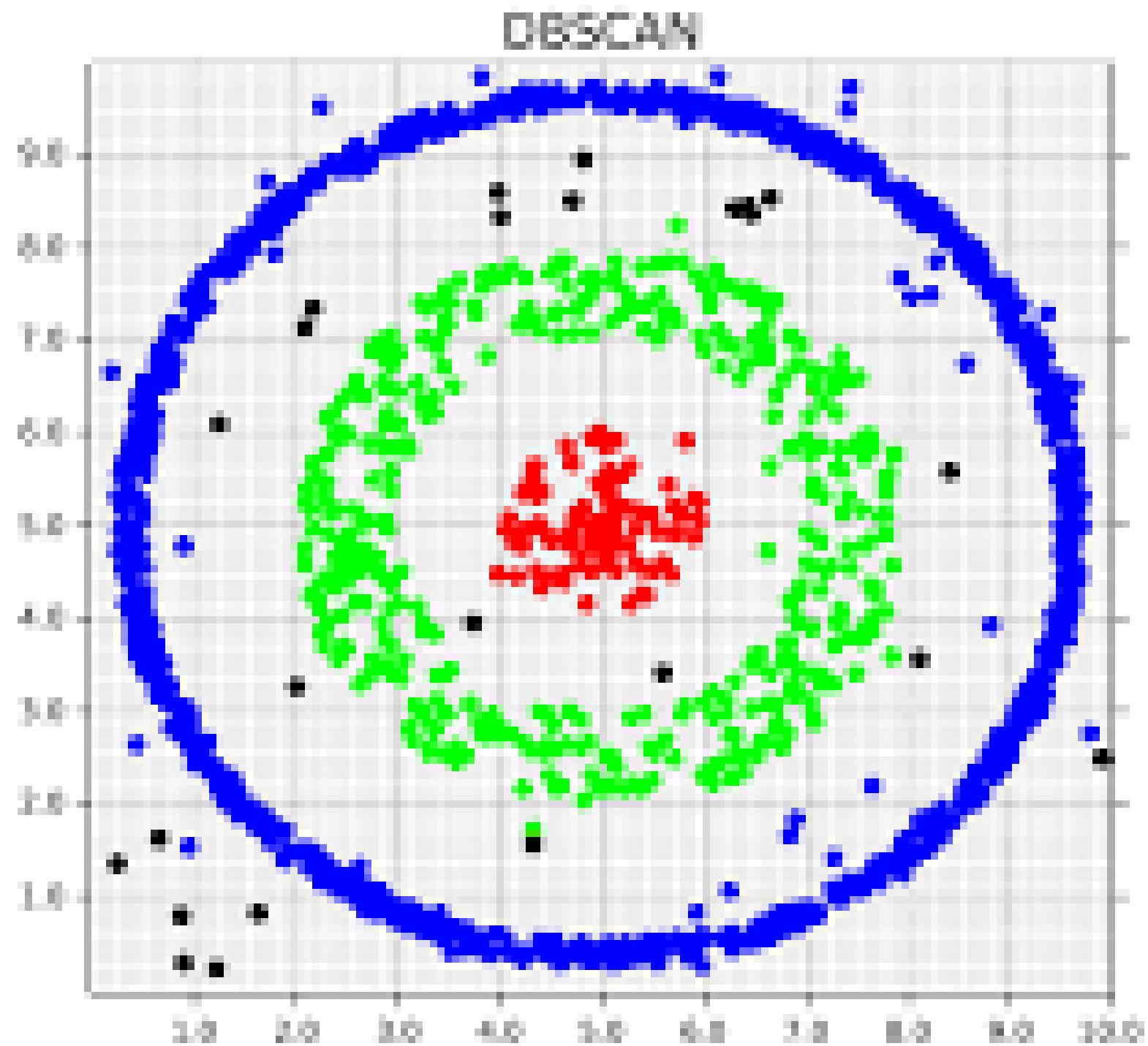
# Clustering for mixed data types

- K-Mode
- K-Prototypes
- K Medoids
  - Gowers distance
    - Manhattan for numeric, Dice for categorical

# K-Means cannot handle asymmetric globular data

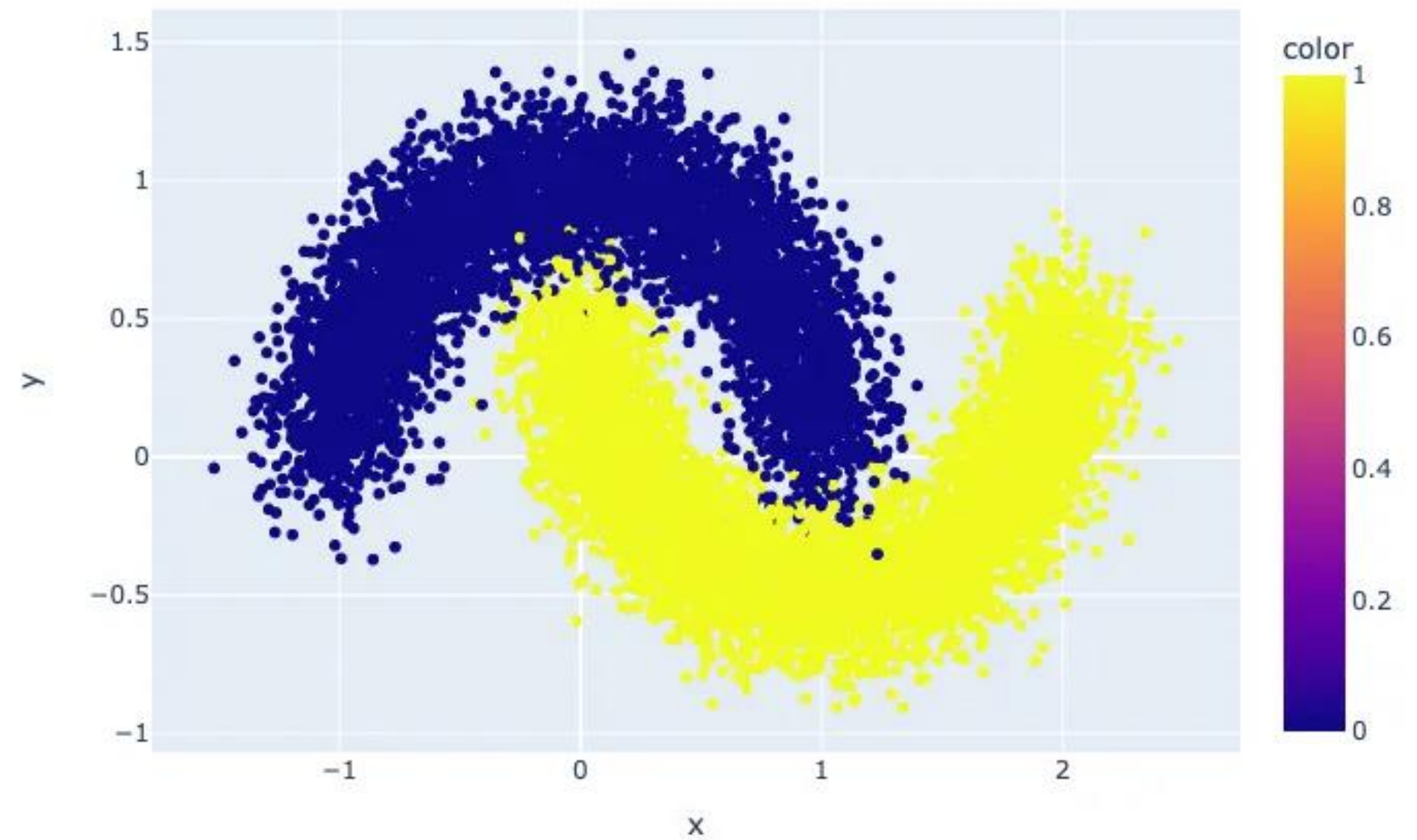
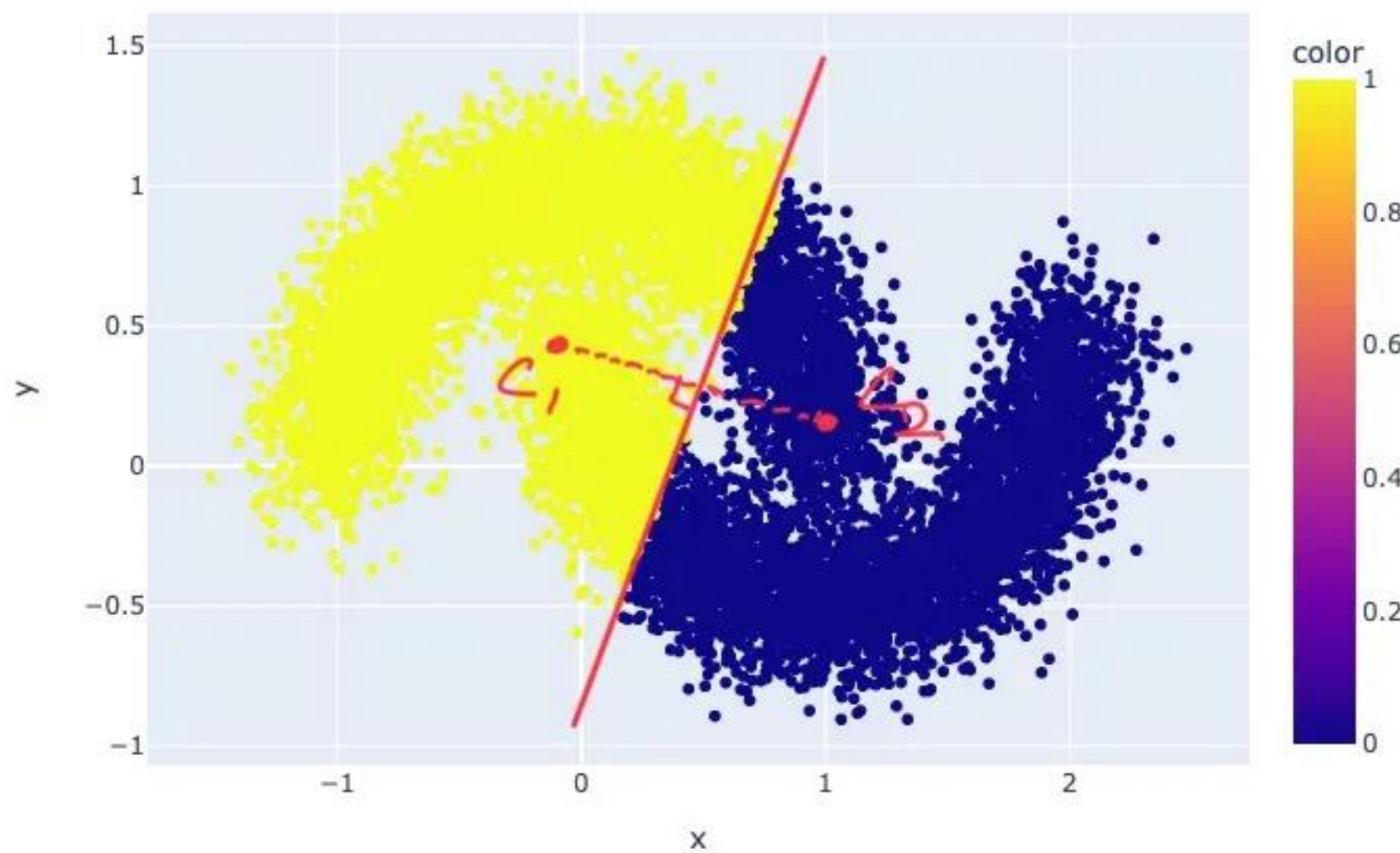


# K-Means cannot handle non globular data



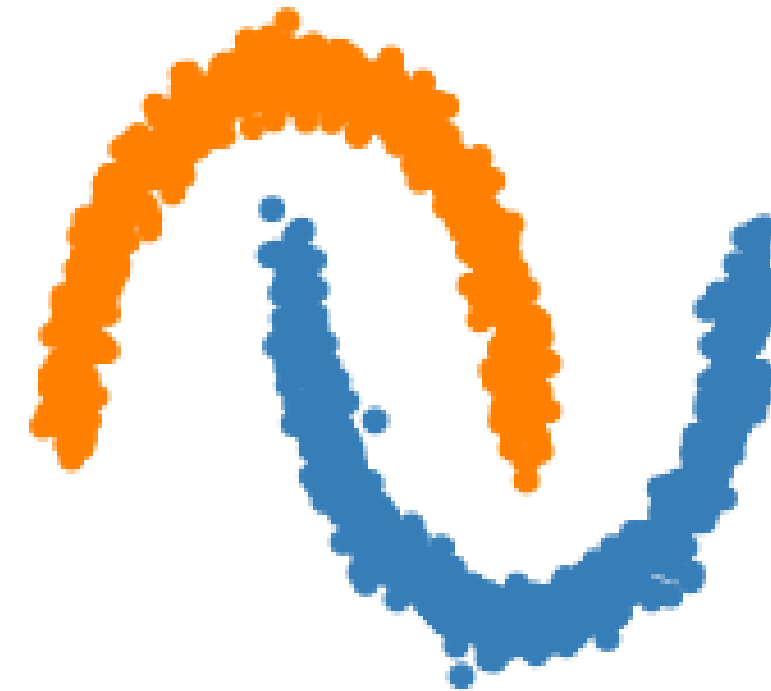
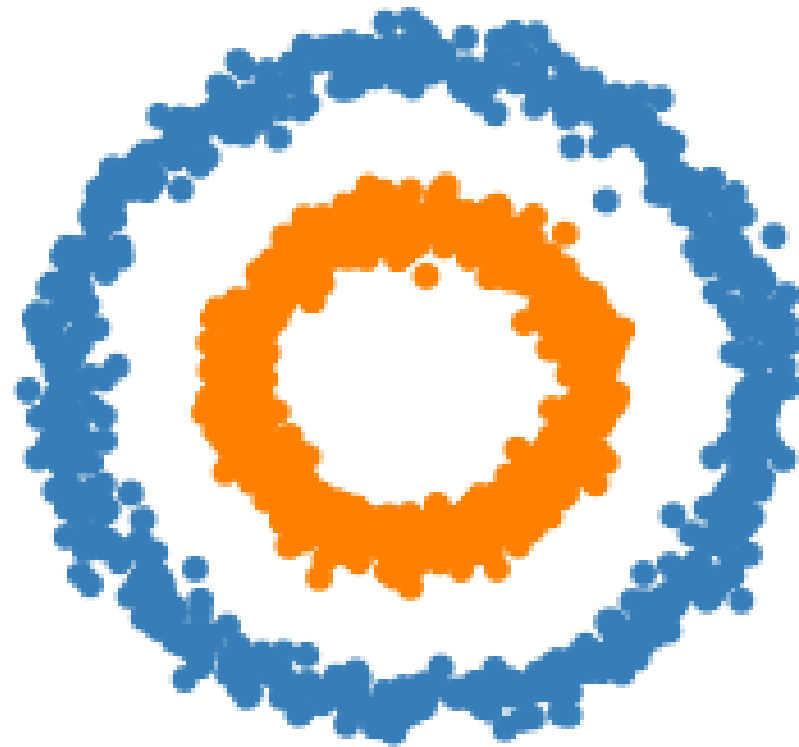


# K-Means cannot handle non globular data

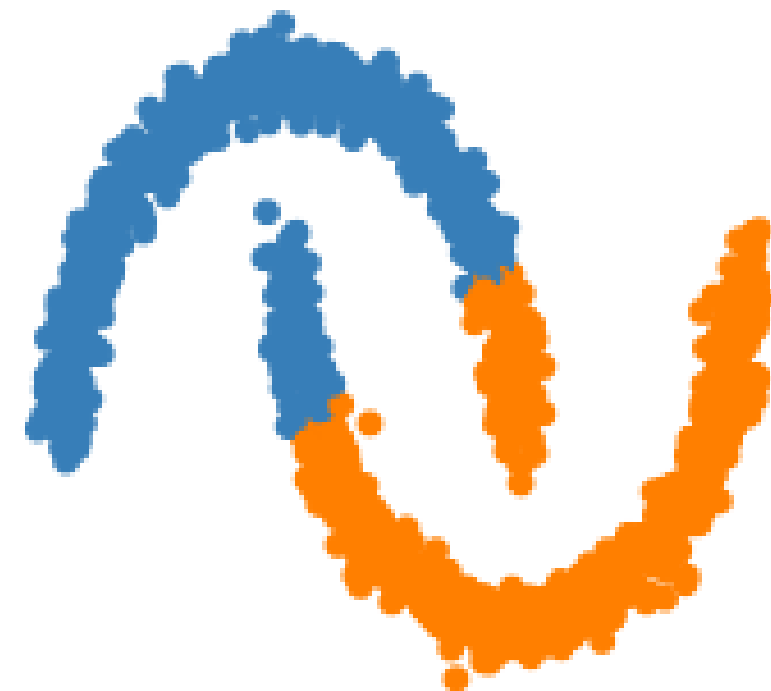


# K-Means cannot handle non globular data

DBSCAN



k-means



# Different types of clustering (optional)

- DBSCAN
  - Some patterns of non globular data, diff metrics
  - NLP clustering, Cosine distance ( $1 - \cos \theta$ )
  - DBSCAN Clearly explained Josh Stammer
    - <https://www.youtube.com/watch?v=RDZUdRSDOok>
  - Evaluation metric is also different (DBCV etc.)
- BIRCH clustering for large datasets
- Mean shift clustering
  - Finds use in unsupervised image segmentation

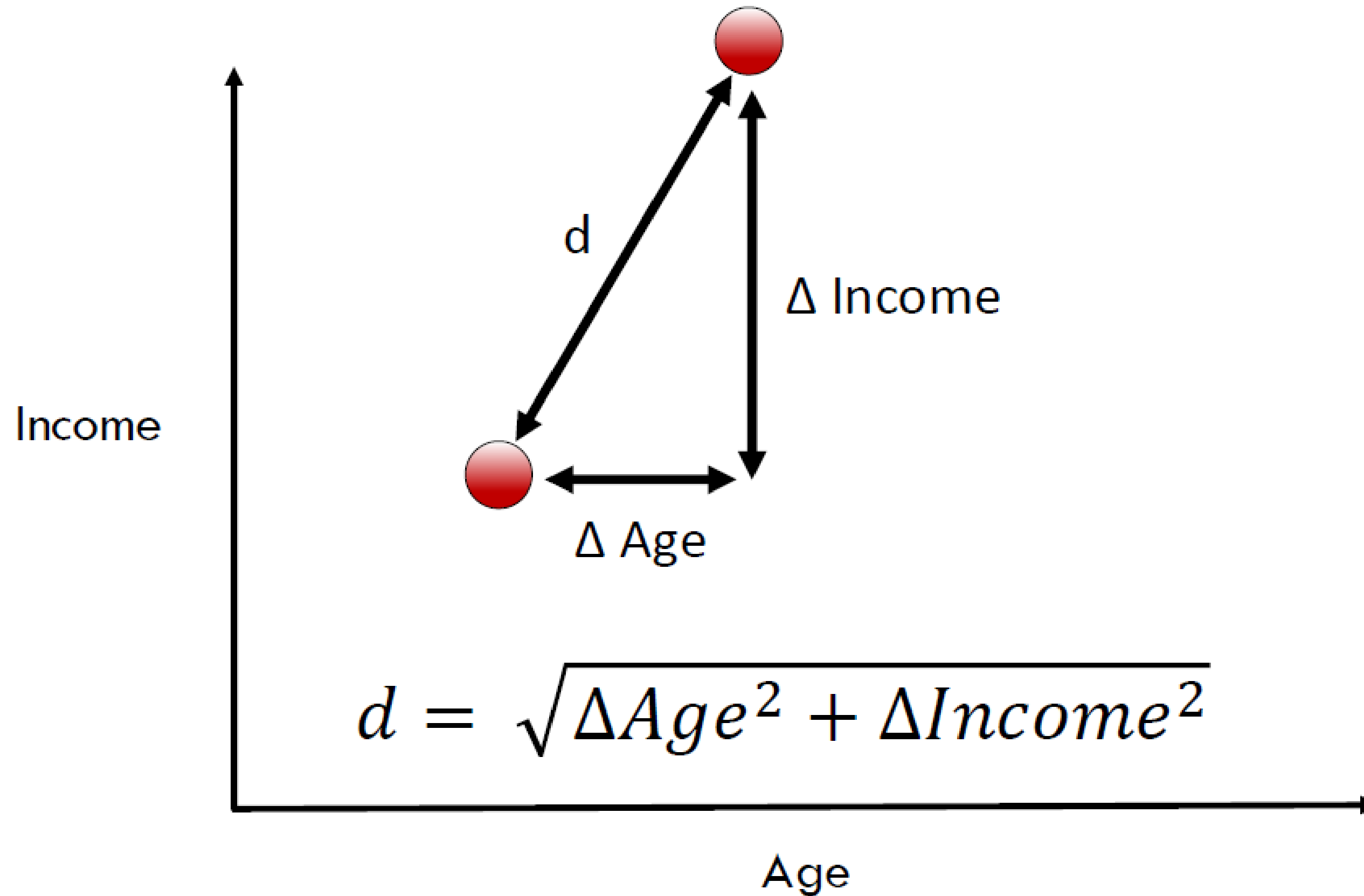


# Case for Distance metrics

- Choice of distance metric is crucial for clustering success
- Also success in any distance based ML method
  - E.g. Nearest Centroid, KNN
- Each metric has strengths and appropriate use cases
  - For e.g. NLP and cosine distance
  - Linear Regression without outliers and Euclidean
- Sometimes metric is chosen by empirical eval (hyperparameter tuning)

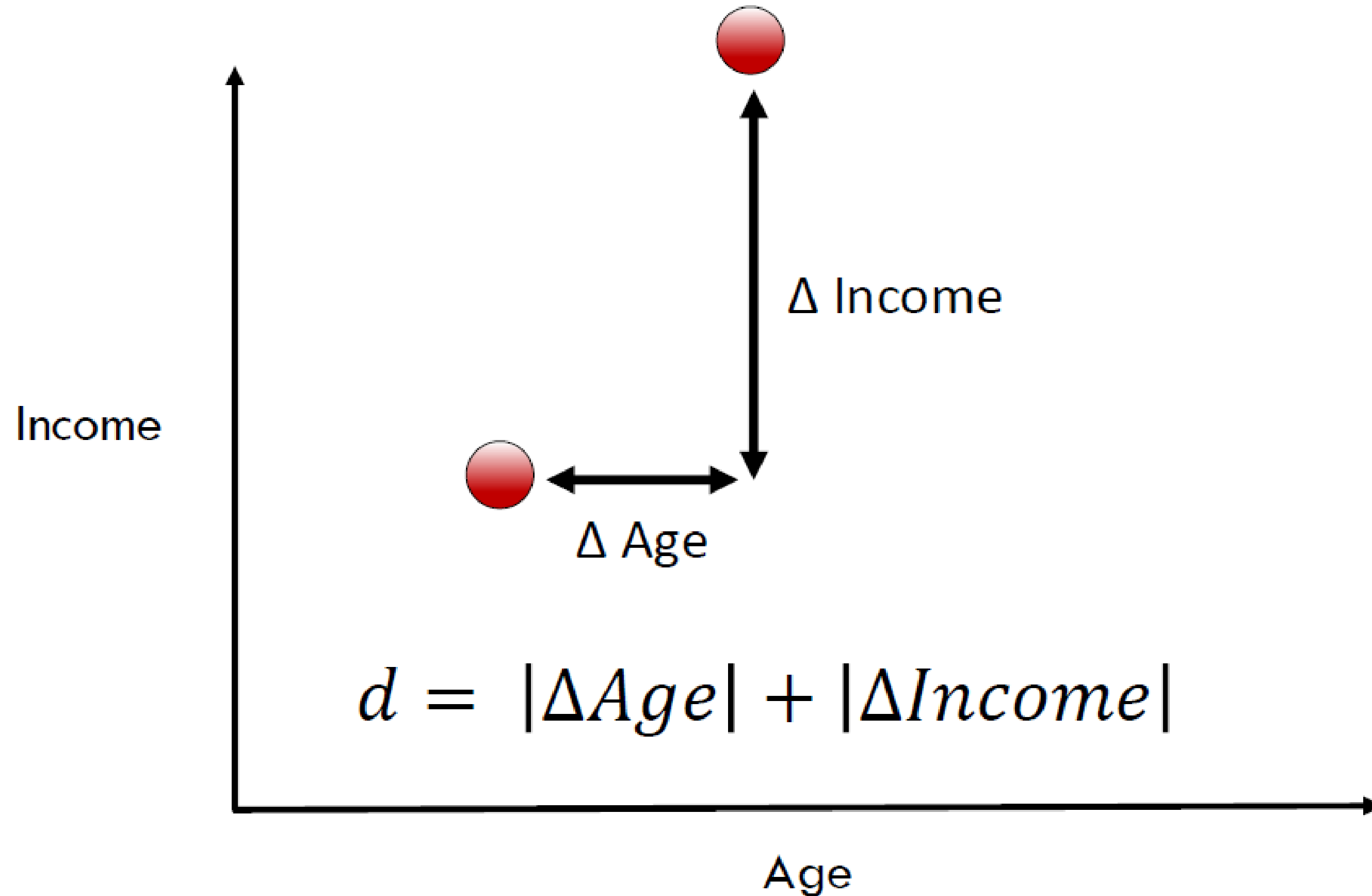


# Euclidean distance





# Manhattan distance



# p norm as generalization of Euclidean

- Numerical attribute: Euclidean distance

$$\mathcal{D}(x, x') = \sqrt{\sum_{i=1}^d (x_i - x'_i)^2}$$

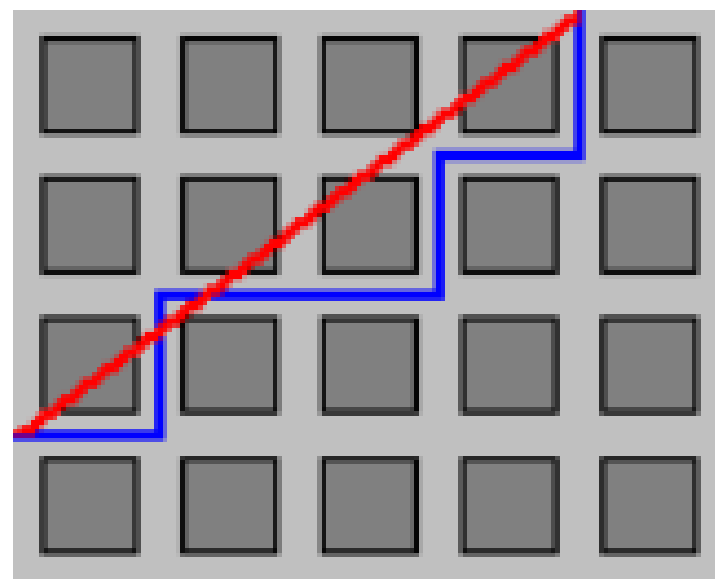
- Pros: Symmetrical, Spherical

- Cons: Sensitive to extreme value of any single attribute

- Minkowski distance (p-norm)

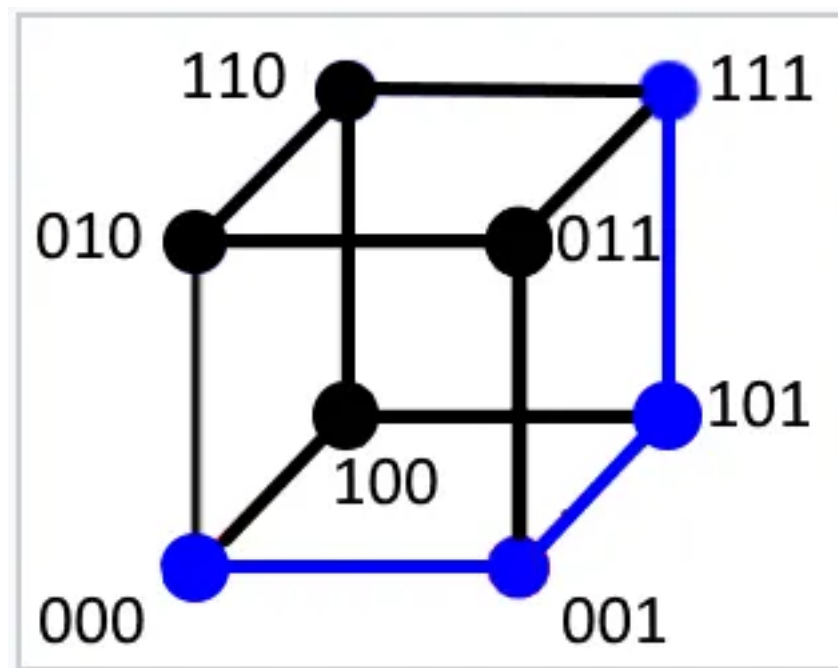
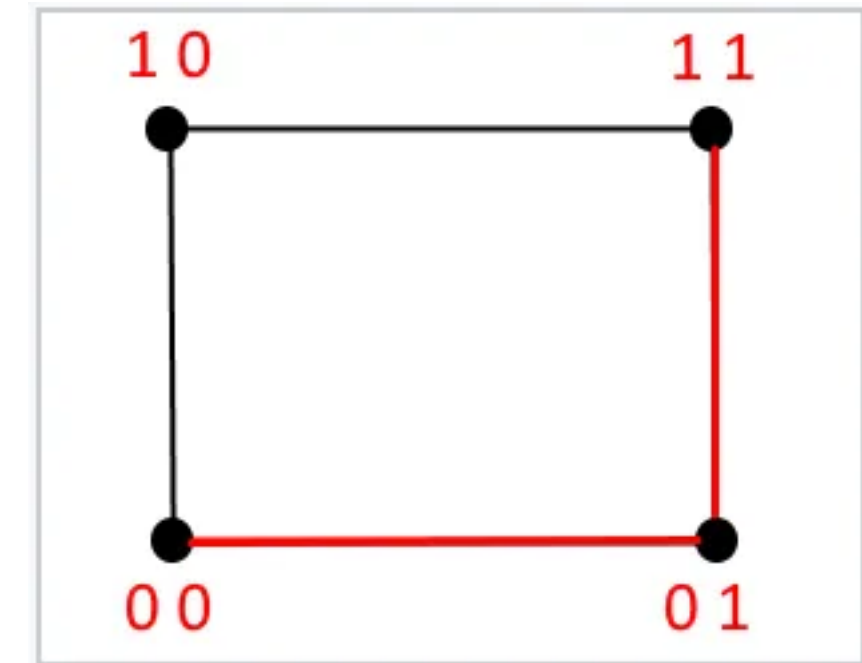
- P=1 Manhattan, p=2 Euclidean

$$\mathcal{D}(x, x') = \sqrt[p]{\sum_{i=1}^d |x_i - x'_i|^p}$$



# Default choice of distance

- Categorical attribute: Hamming distance
  - Number of attributes where  $x, x'$  differ
  - Best for categorical attributes



Special case of One Hot encoded feature

$$\mathcal{D}(x, x') = \sum_{i=1}^d \mathbb{I}_{x_i = x'_i}$$

$$\mathcal{D}(x, x') = \sum_{i=1}^d |x_i - x'_i|$$

Category	F1	F2	F3
Low	0	0	1
Medium	0	1	0
High	1	0	0



# p norm for different p

- Minkowski distance (p-norm)  $p=0$

- When  $x$  not equal to  $x'$

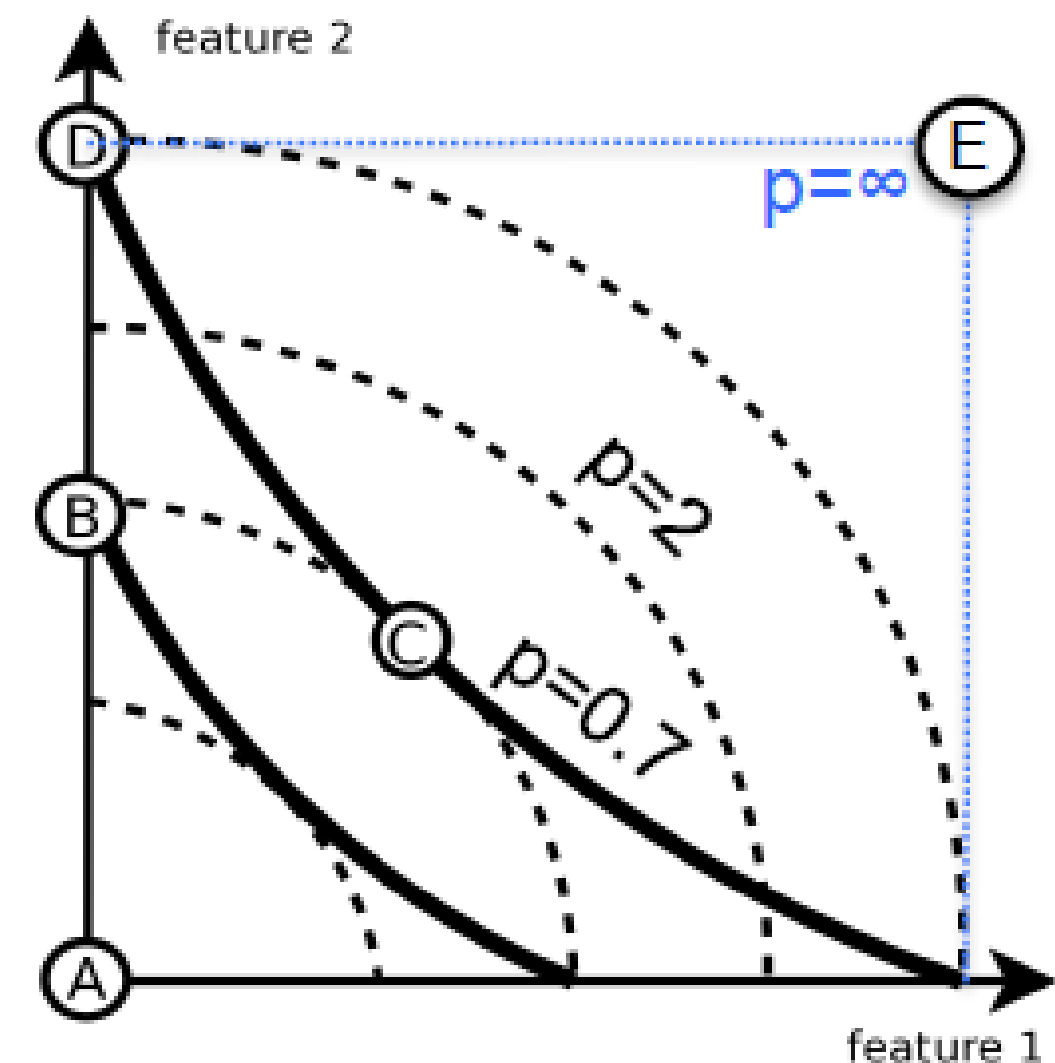
- When  $x$  equal to  $x'$

- Behaves like Hamming distance

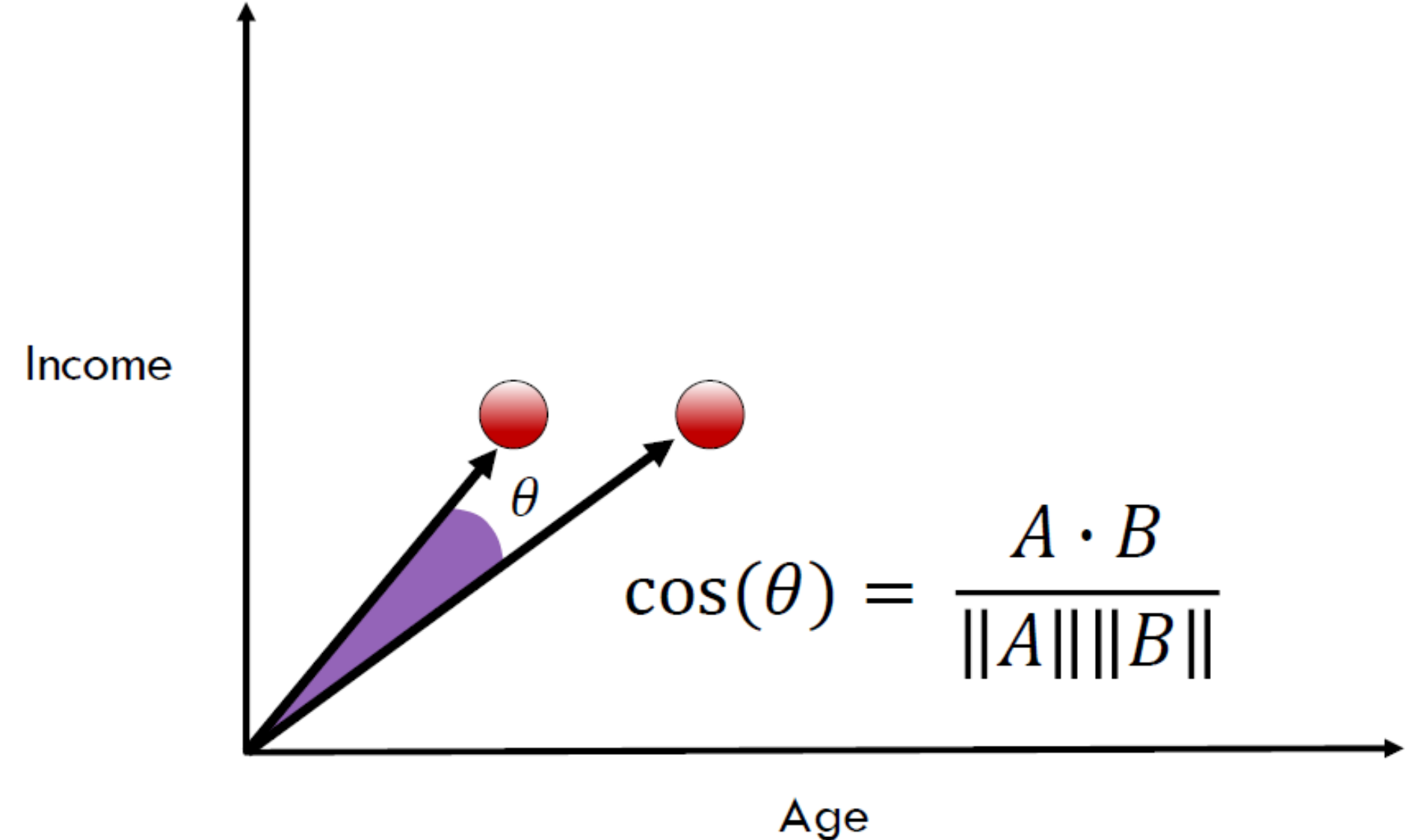
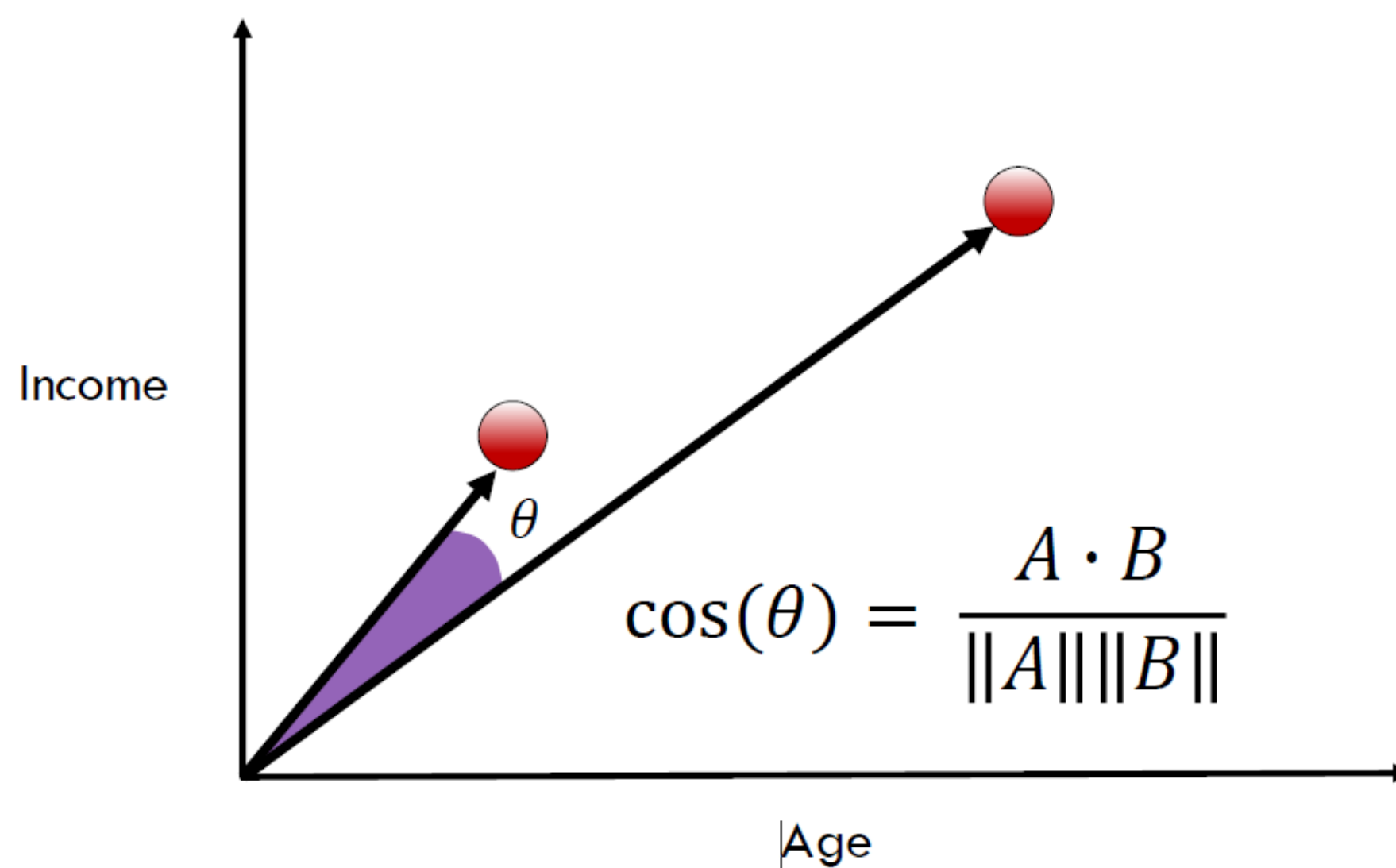
- $p=\text{infinity}$       $\mathcal{D}(x, x') = \max_d |x_i - x'_i|$

- Different  $p$

$$\mathcal{D}(x, x') = \sqrt[p]{\sum_{i=1}^d |x_i - x'_i|^p}$$



# Cosine similarity & Cosine distance



- Euclidean distance does not matter
- Angular separation matters (e.g. NLP)
- As  $\theta$  decreases Cosine similarity increases
- Cosine distance  $= 1 - \cos\theta$

# Jaccard similarity & Jaccard distance

- Applies to set similarity, community similarity

**Sentence A:** “I like chocolate ice cream.”

set A = {I, like, **chocolate**, ice, **cream**}

**Sentence B:** “Do I want chocolate cream or vanilla cream?”

set B = {Do, I, want, **chocolate**, **cream**, or, vanilla}

- Jaccard similarity  $= \frac{A \cap B}{A \cup B} = \frac{3}{9}$
- Jaccard distance  $= 1 - \frac{A \cap B}{A \cup B} = 1 - \frac{3}{9} = \frac{6}{9}$



# Custom distance metrics

- Always non negative  $\|x_1 - x_2\| \geq 0$
- $\|x_1 - x_2\| = 0 \iff x_1 = x_2$
- $\|x_1 - x_2\| = \|x_2 - x_1\|$
- Triangle inequality  $\|x + y\| \leq \|x\| + \|y\|$
- Should satisfy all of the above (generally speaking)

**Intuitively  
true:  
Sides of  
triangle**

# Distance metrics in sklearn

- `from sklearn.metrics import pairwise_distances`
- `dist = pairwise_distances(X, X', metric='')`
- `metric = euclidean, manhattan, cosine, jaccard`

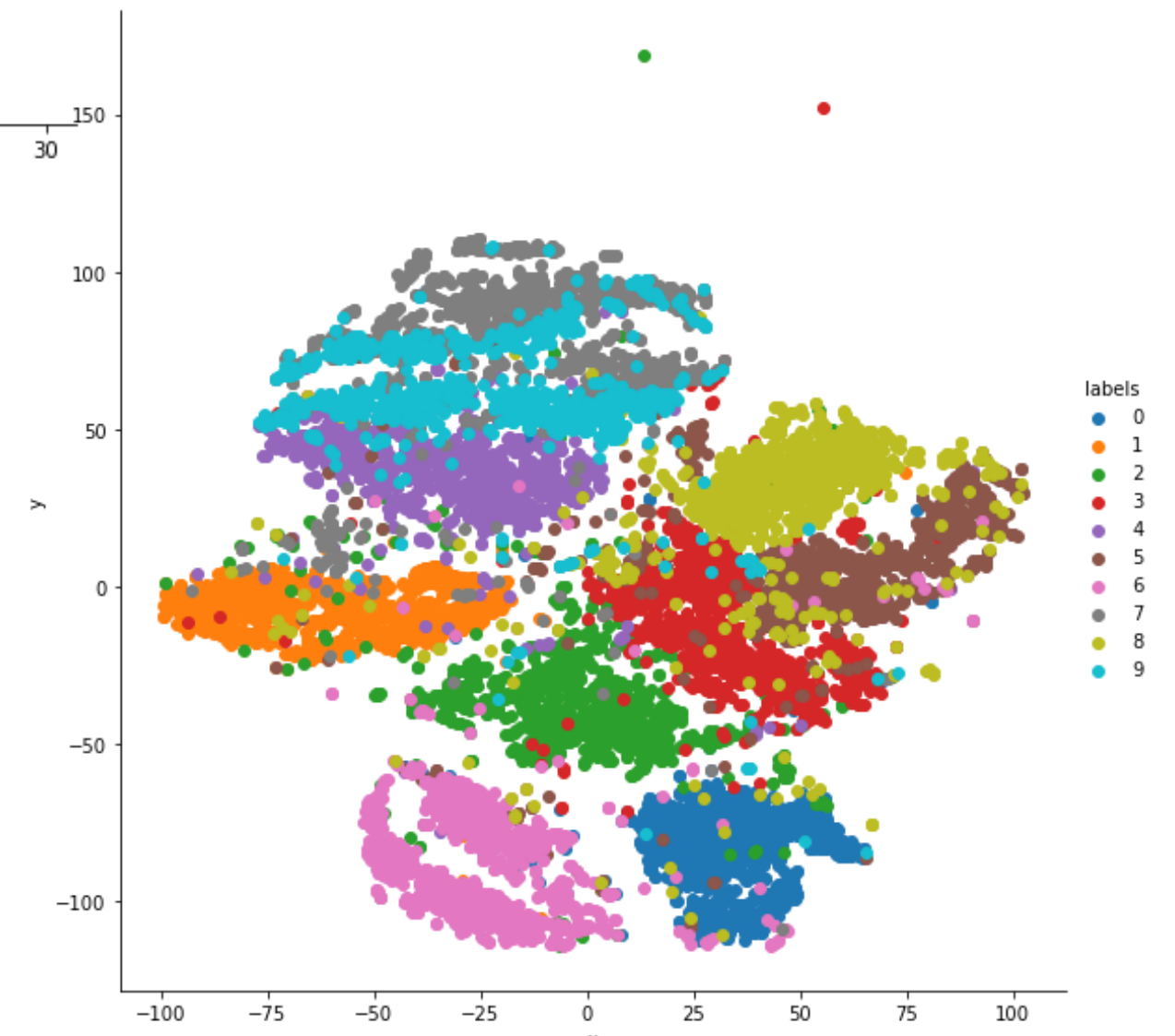
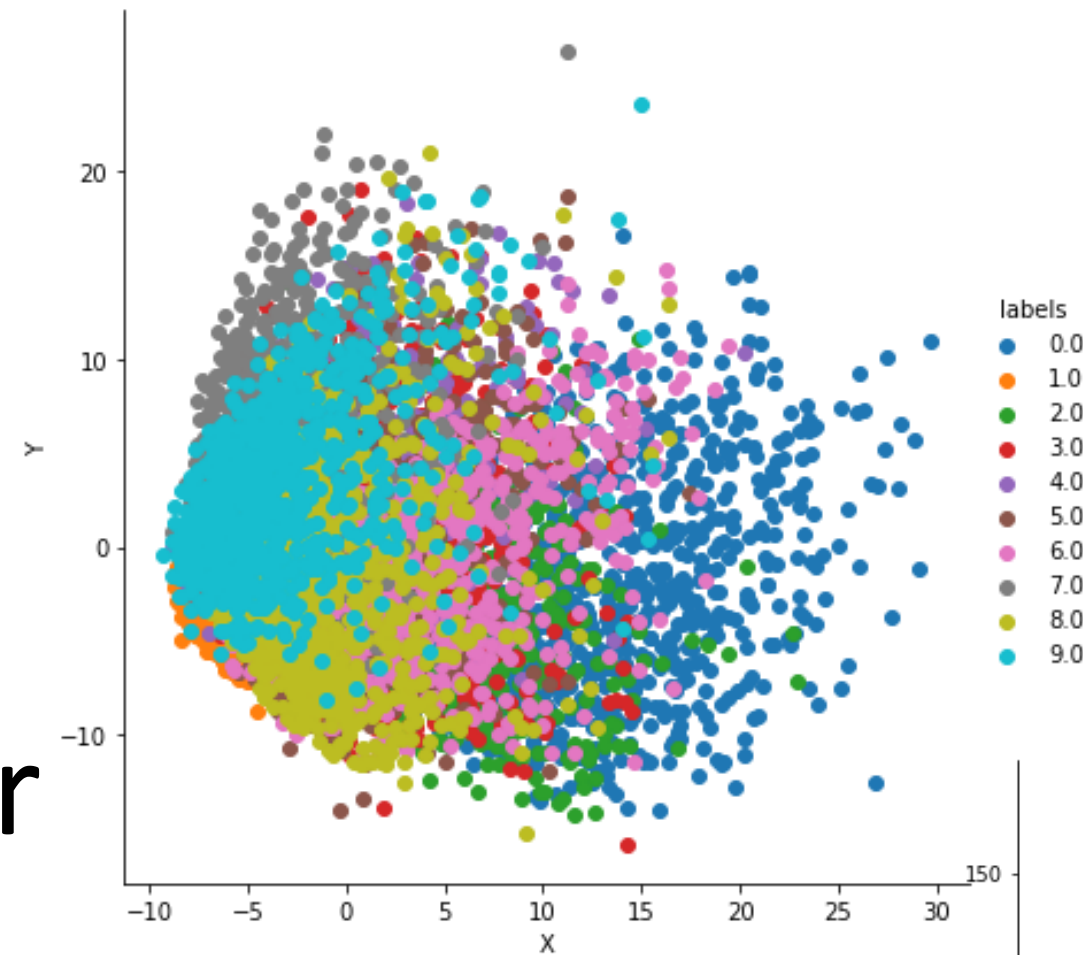


# Dimensionality Reduction + Clustering for visualization (optional)



# Dimensionality reduction

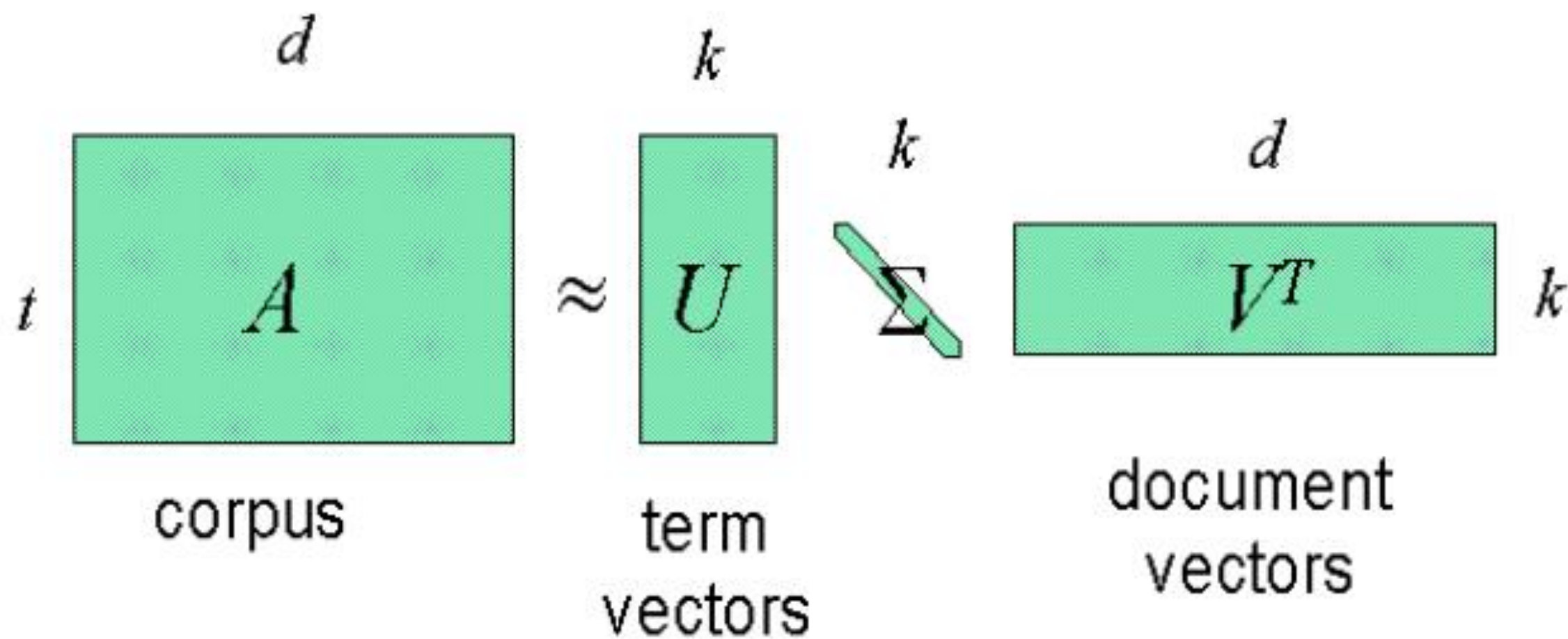
- Apply dimensionality reduction
  - PCA
  - T-SNE
- Then clustering in 2-D or 3-D
- E.g. PCA and t-SNE with MNIST for 2D clustering
  - Notice t-SNE gives better cluster separation





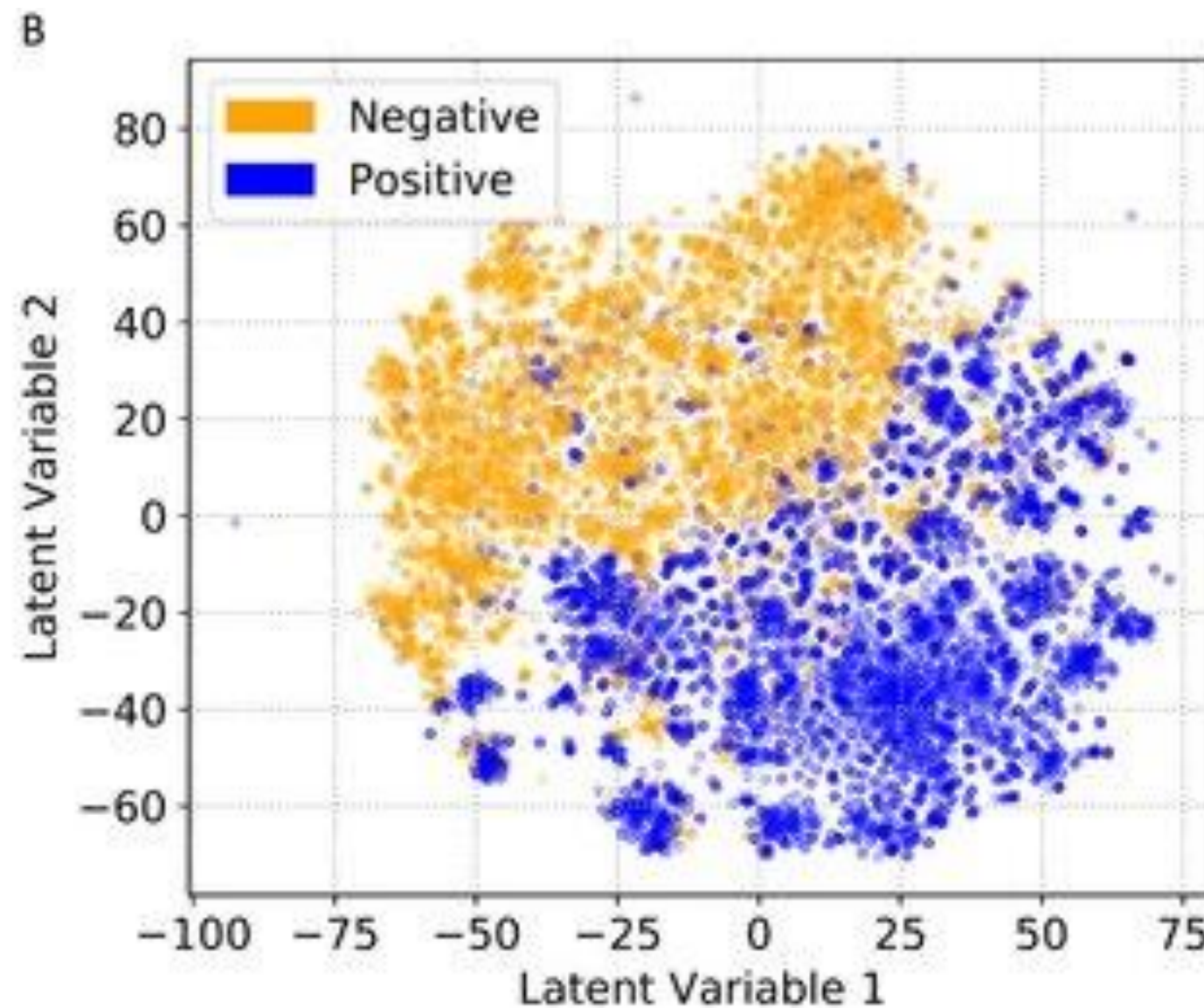
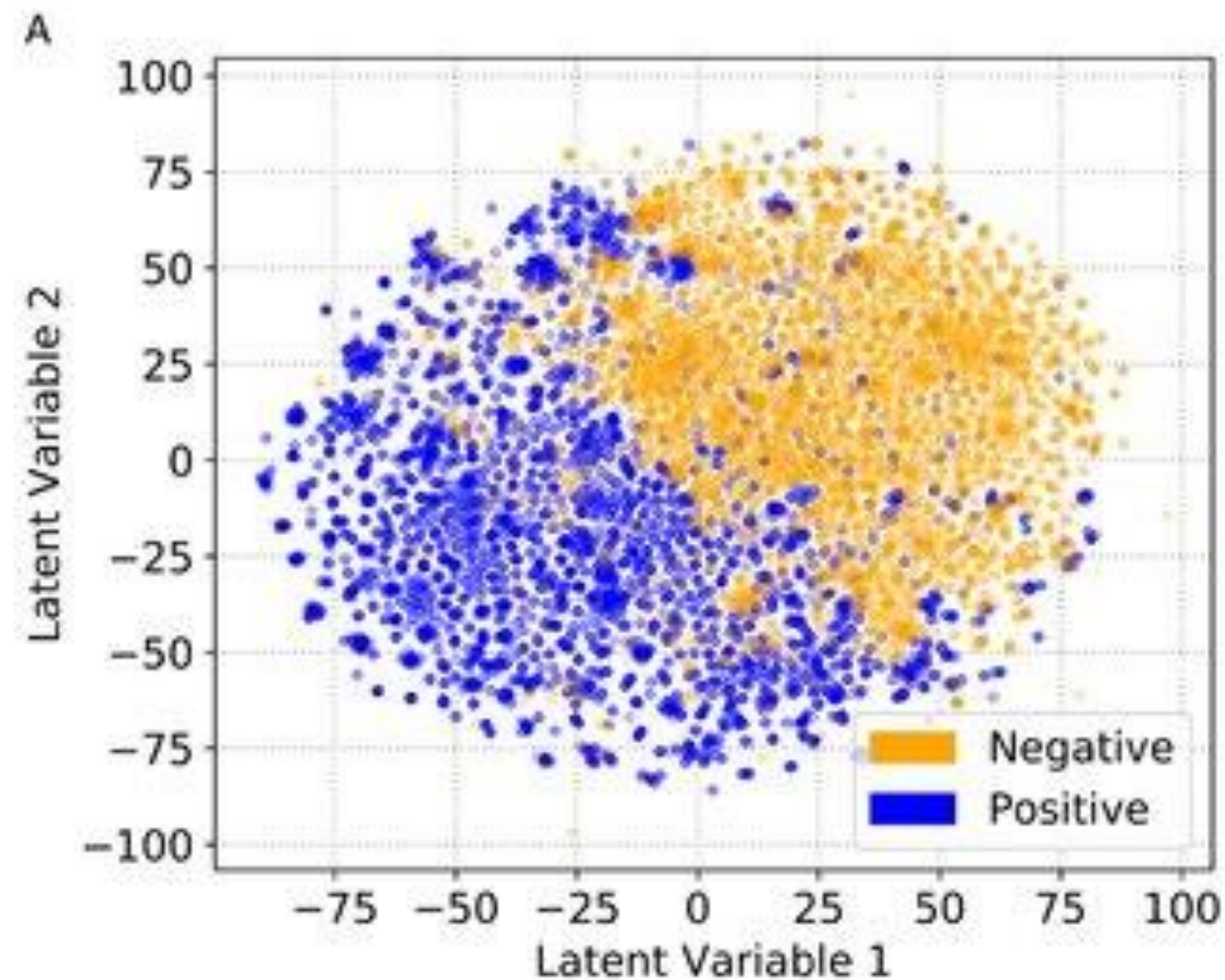
# Dimensionality reduction

- Truncated SVD
  - Sparse Matrix cases (such as TF-IDF)



# Clustering after dim reduction

- Truncated SVD
  - Sparse Matrix cases (such as TF-IDF)







QUESTIONS