Recap

- Curve fitting is logical way of looking at regression problems
- Line, plane & hyperplane - one form $y = w^T x + b$
- Learning algorithm & hypothesis function
- How many degrees is just right
- Underfitting, Overfitting and just right fit
- Bias Variance Tradeoff
- Overfitting discovered by Cross validation and error increasing beyond a point
- Overfitting learns noise

Geometric Perspective of Classification

Classification: Diabetic dataset

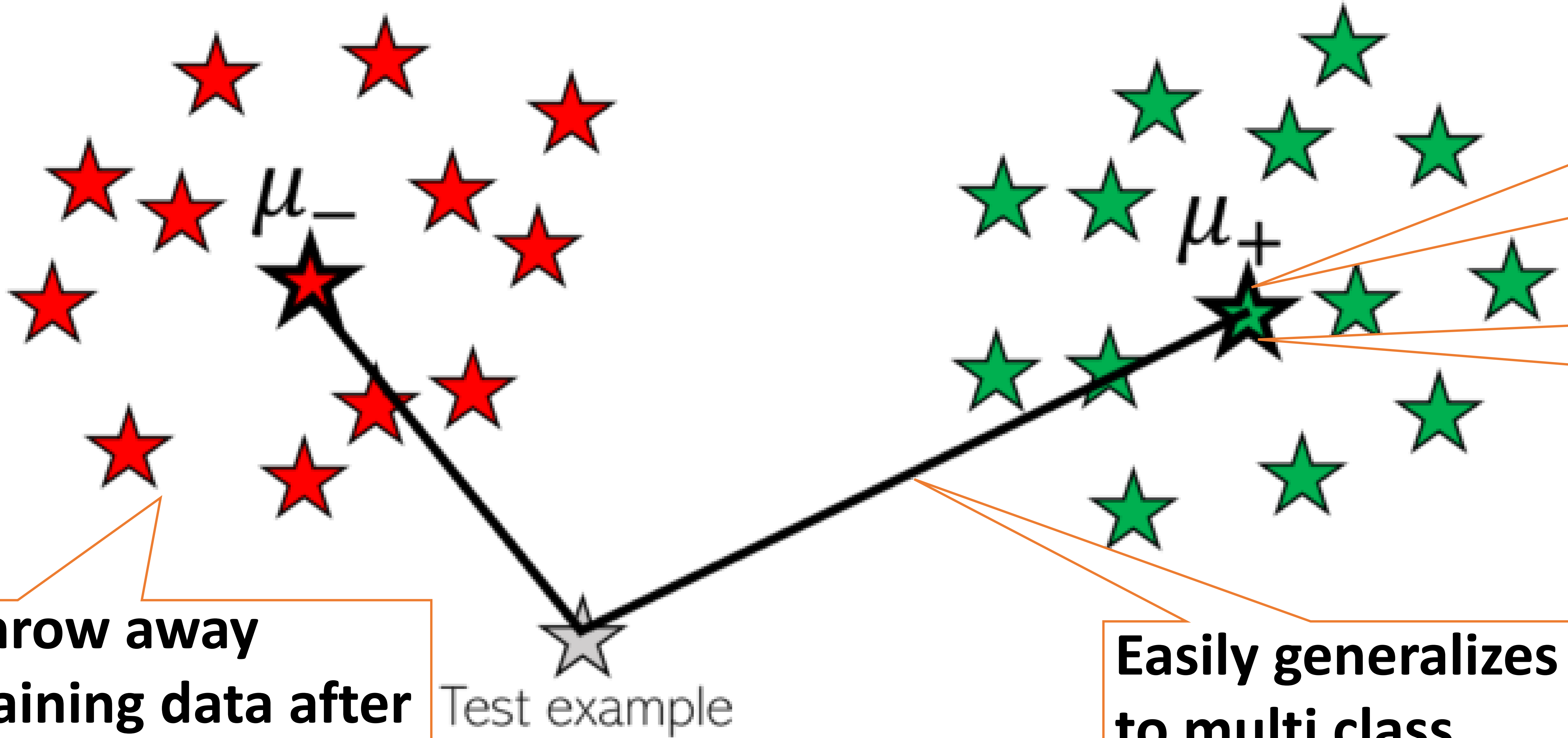
Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1

- Geometric perspective is the most intuitive way of looking at classification problems
- Classic decision boundary in linear form $w^T x + b = 0$

Nearest Centroid

Called Learning with Prototype (LwP)

- Binary classification
  - Given height and weight predict South Asian or not



Represent each class with a prototype vector

Mean is called class prototype

Throw away training data after learning centroid

Easily generalizes to multi class classification

How to calculate mean vector?

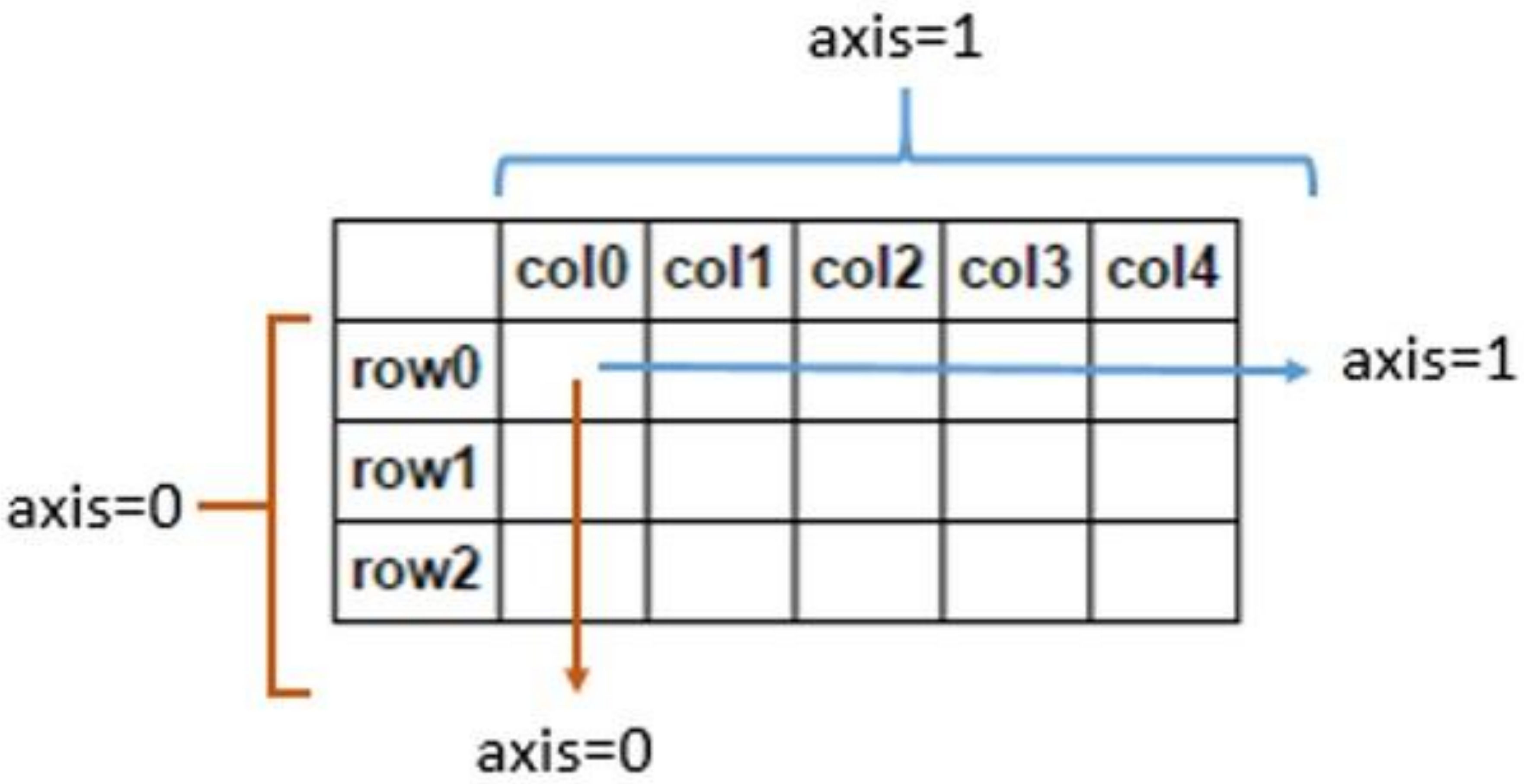
	HR	BP	Temp
Patient-1	76	126	38.0
Patient-2	74	120	38.0
Patient-3	72	118	37.5
Patient-4	78	136	37.0

$$x^{(1)} = \begin{bmatrix} 76 \\ 126 \\ 38 \end{bmatrix} \qquad x^{(2)} = \begin{bmatrix} 74 \\ 120 \\ 38 \end{bmatrix}$$

$$\bar{x} = \frac{1}{4}\left(x^{(1)} + x^{(2)} + x^{(3)} + x^{(4)}\right) \qquad \bar{x} = \begin{bmatrix} 75 \\ 125 \\ 37.625 \end{bmatrix}$$

How to calculate mean vector?

	HR	BP	Temp
Patient-1	76	126	38.0
Patient-2	74	120	38.0
Patient-3	72	118	37.5
Patient-4	78	136	37.0



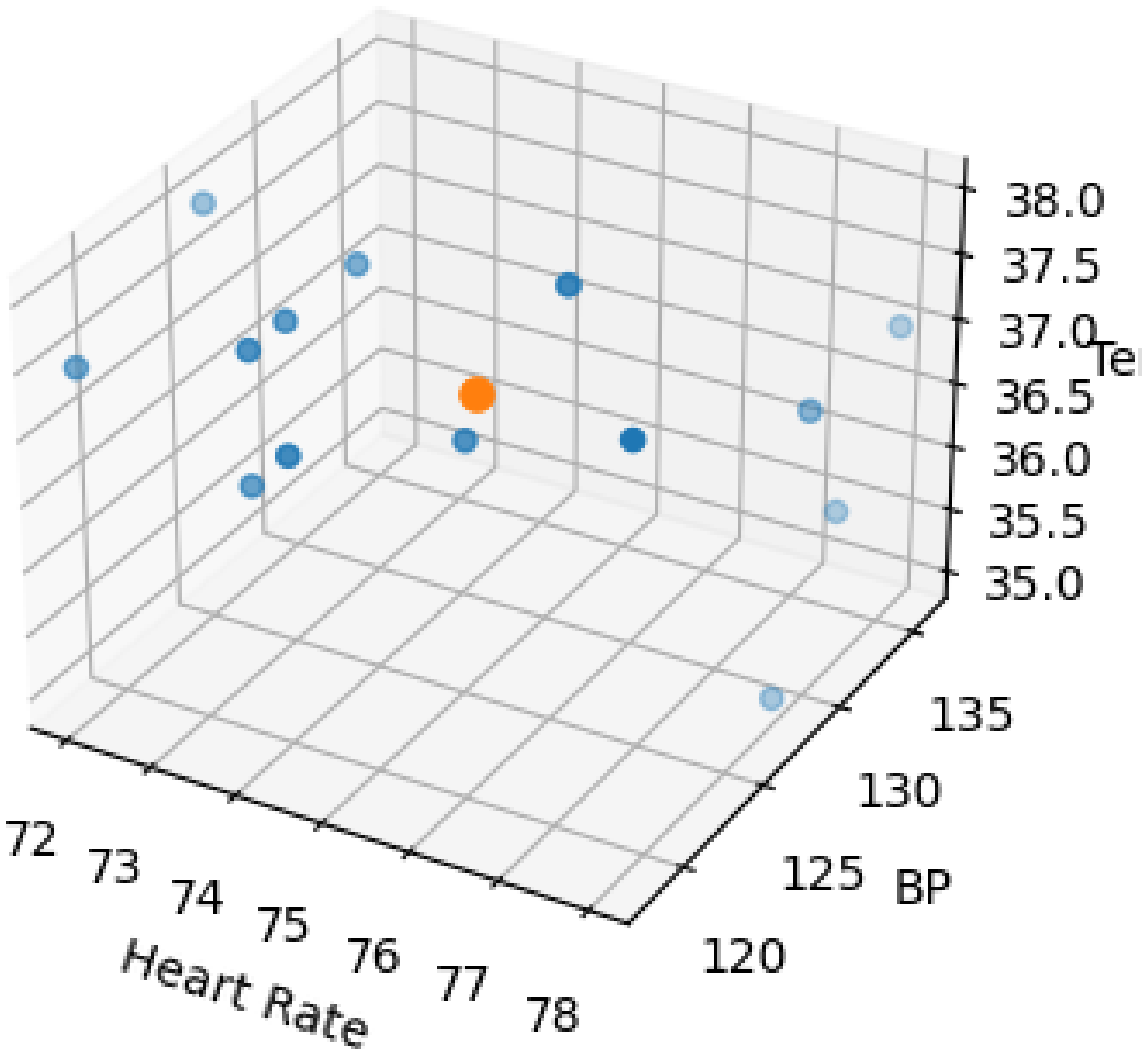

```
avg_patient = (1/patients.shape[0]) * np.sum(patients, axis=0)
```

```
avg_patient = np.mean(patients, axis=0)
```

Mean vector is centroid of dataset

	HR	BP	Temp
Patient-1	76	126	38.0
Patient-2	74	120	38.0
Patient-3	72	118	37.5
Patient-4	78	136	37.0

$$\bar{x} = \begin{bmatrix} 75 \\ 125 \\ 37.625 \end{bmatrix}$$



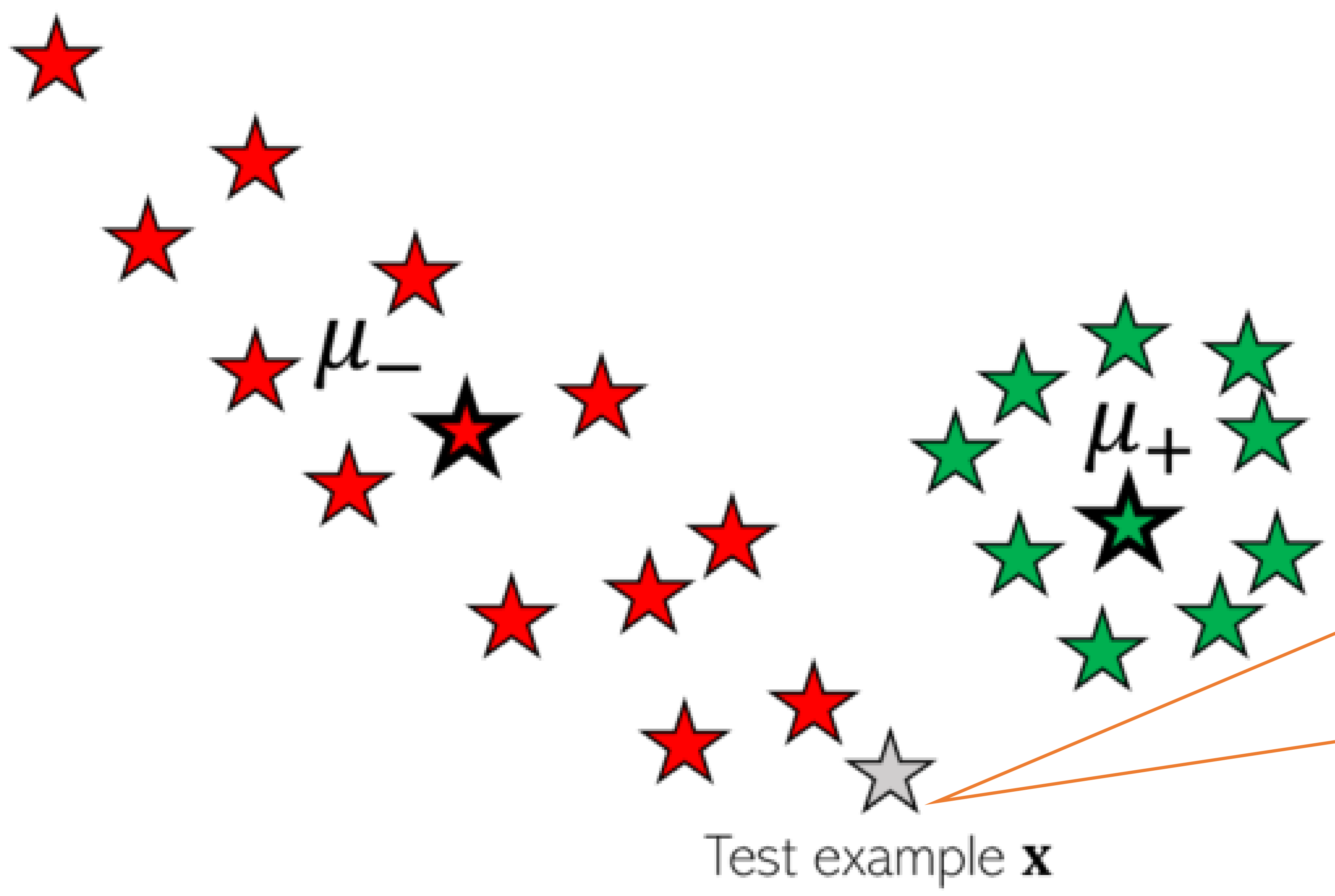
How to find the distance between vectors?

- https://www.geogebra.org/calculator/qmayzjm9

Nearest Centroid observations

- Model based (Centroids and class mappings are the model), parametric
- Pros: Fast and easy
- Cons: Primitive Classifier
- Not really a machine learning model as such
- Just statistical data analysis & a simple pattern usage
- What is there to “LEARN”?
    - Weighted distances can be learnt (more on this later)

Nearest Centroid – Failure cases



Fix: Can use any of

a. Feature scaling (FS)

b. Weighted distance (such as Mahalanobis distance)

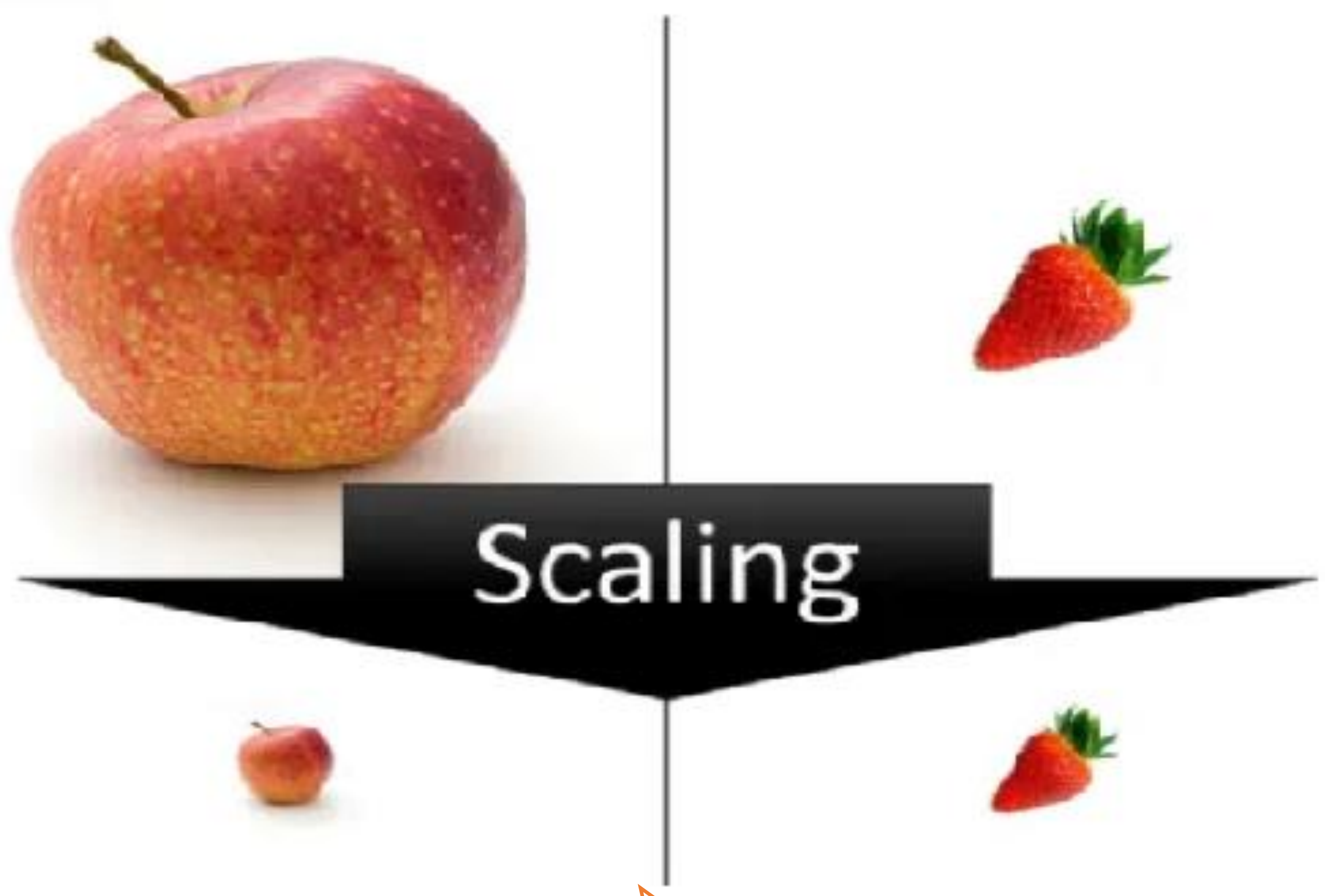
c. Probability distribution

- Test e.g. x logically belongs to red class (Euclidean)
- Nearest Centroid predicts green class

Feature Scaling

Feature Scaling

- Important data pre-processing step
- Prevents feature with large values dominate training
- Features should not dominate by
  - Units
  - Sheer variance in different scale
- Features should dominate by coefficient of variation $\frac{\sigma}{\mu}$



NOTE: If scaling is applied at training time, then scaling should be applied on test data during prediction also

Feature Scaling (contd.)

- All distance based algorithms are sensitive to feature scale
  - Linear Regression to Neural Networks
  - Everything in between
- Information Theory based algorithms are insensitive
  - Decision Tree, Random Forest

z-transformation

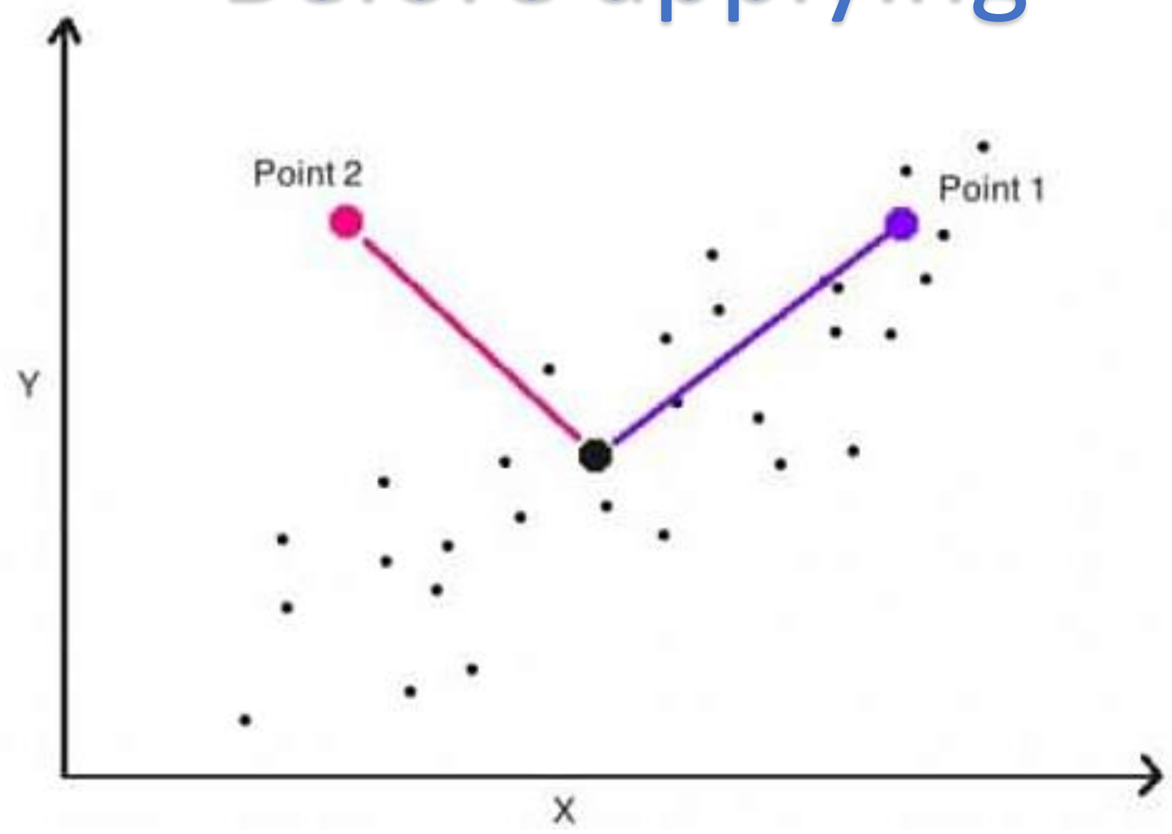
Also called Standardization

$$z = \frac{x - \mu}{\sigma}$$

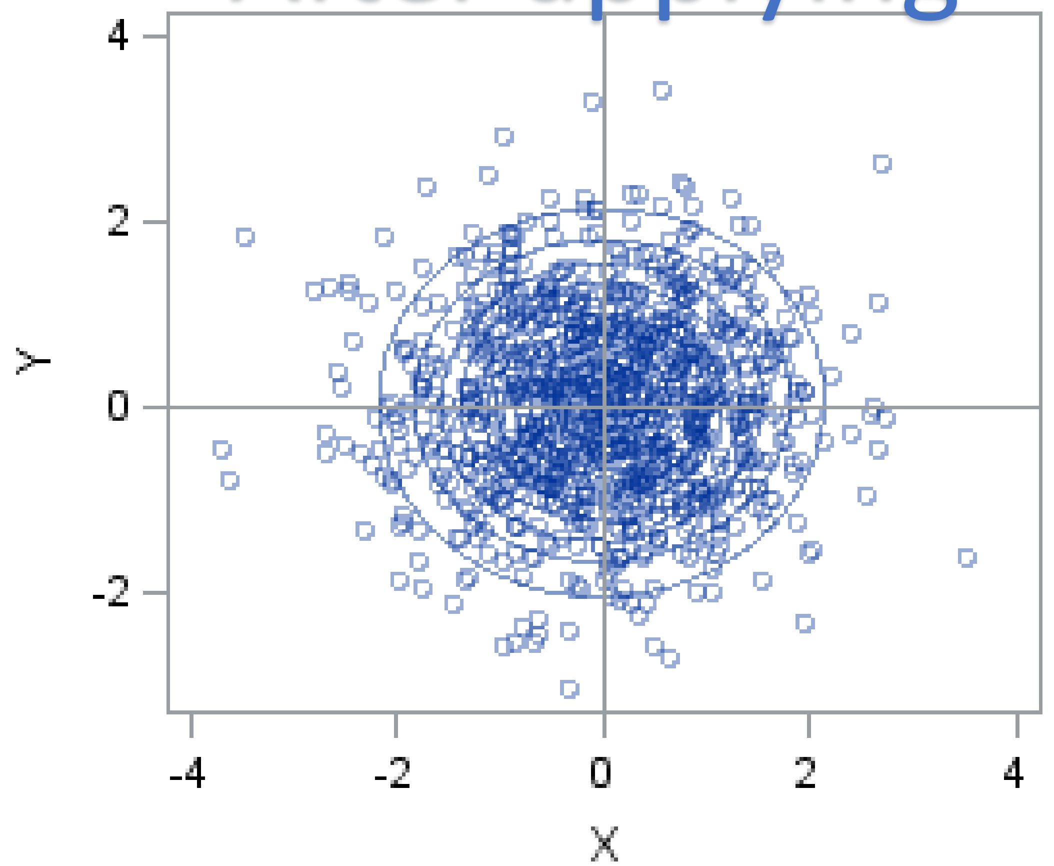
Available in sklearn as StandardScaler

- Feature vector is mean centered $x - \mu$
  - Makes the feature zero centered
- Scale feature vector by reciprocal of standard deviation $\frac{1}{\sigma}$

Before applying

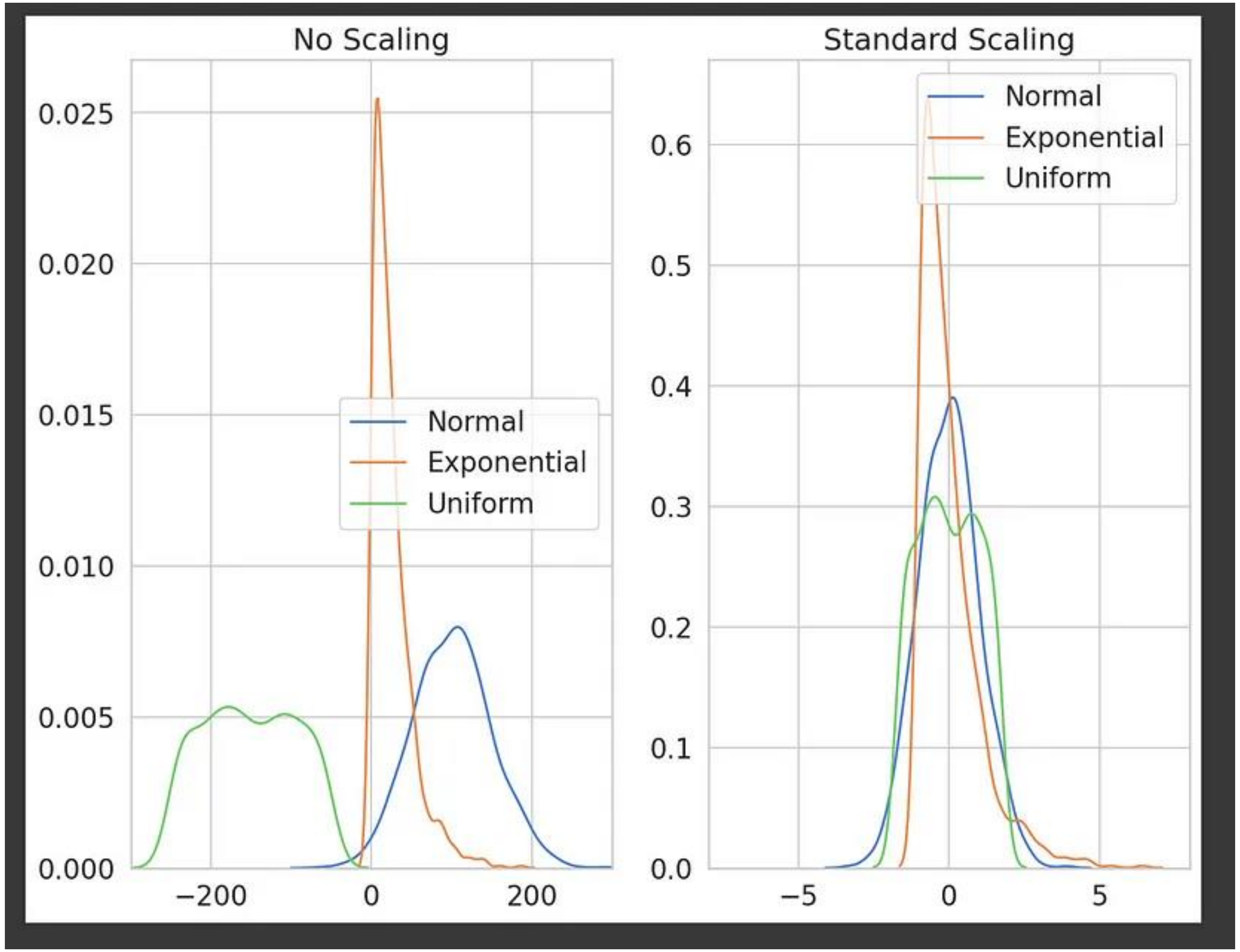


After applying



z-transform (contd.)

- Z-transform does not change the distribution
- Only zero centers and rescales
- Can be applied to non-gaussian data
- Makes the data unit-less
- Mean = 0
- Variance = 1



Other transformations

- Major ones (to be covered later)
  - Min-Max Normalization
  - Robust Scaling
  - Log Transformation
  - Power Transformation
- Selected based on characteristics of data and end goal
- Chain transformations

Three confusing terms

- Feature Selection, Feature Engineering, Feature Extraction
- Feature Selection: Select raw features based on importance
- Feature Engineering: Convert raw features into refined features to improve prediction
  - E.g.: Transformations
- Feature Extraction: Create synthetic features by combining raw/engineered features
  - Principal Component Analysis (PCA)

Weighted distances

Nearest Centroid – Weighted distance

- n is number of features $a, b \in \mathcal{R}^n$

$$d_2(\mathbf{a}, \mathbf{b}) = \|a - b\|_2 = \sqrt{\Sigma_{i=1}^n (a_i - b_i)^2}$$

$$= \sqrt{(a-b)^T(a-b)} = \sqrt{a^Ta + b^Tb - 2a^Tb}$$

- $d_2(\mathbf{a}, \mathbf{b}) = \|a - b\|_w = \sqrt{\Sigma_{i=1}^n w_i (a_i - b_i)^2}$

$$= \sqrt{(a-b)^T W (a-b)}$$

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & w_{n-1} & 0 \\ 0 & 0 & 0 & w_n \end{bmatrix}$$

Diagonal matrix n x n

w can be learnt

Mahalanobis distance

- $$d_2(\mathbf{a}, \mathbf{b}) = \sqrt{(a-b)^T W (a-b)}$$

$$W = \Sigma^{-1}$$

n x n any symmetric matrix then Mahalanobis-like distance

Other symmetric matrices can be learnt

Σ is covariance matrix, then Mahalanobis distance

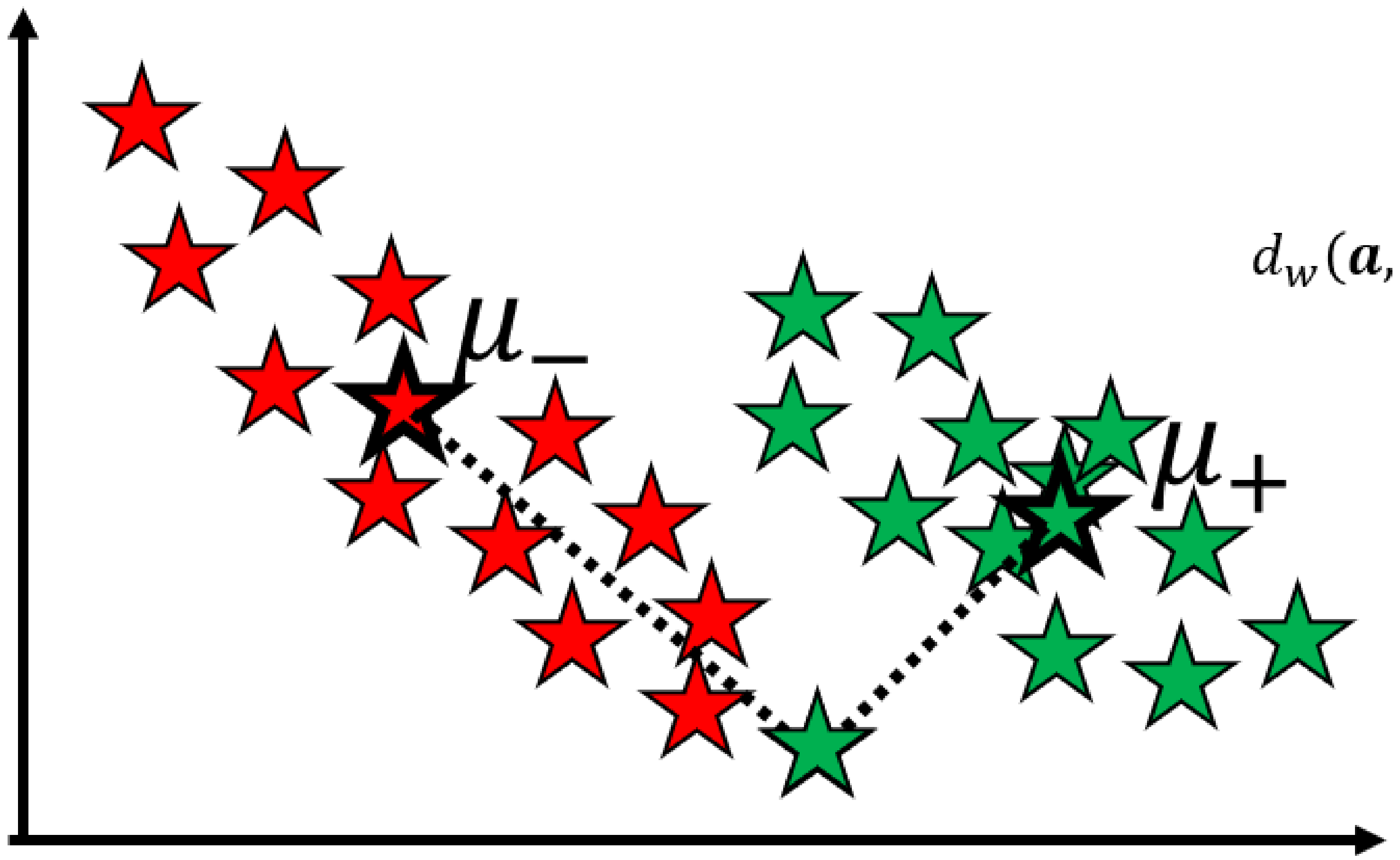
Covariance matrix is symmetric.
Inverse also symmetric

$$\Sigma = \begin{bmatrix} \sigma_1^2 & Cov_{12} & \dots & Cov_{1n} \\ Cov_{21} & \sigma_2^2 & \dots & Cov_{2n} \\ .. & .. & .. & .. \\ Cov_{(n-1)1} & .. & \sigma_{n-1}^2 & Cov_{(n-1)n} \\ Cov_{n1} & Cov_{n2} & .. & \sigma_n^2 \end{bmatrix}$$

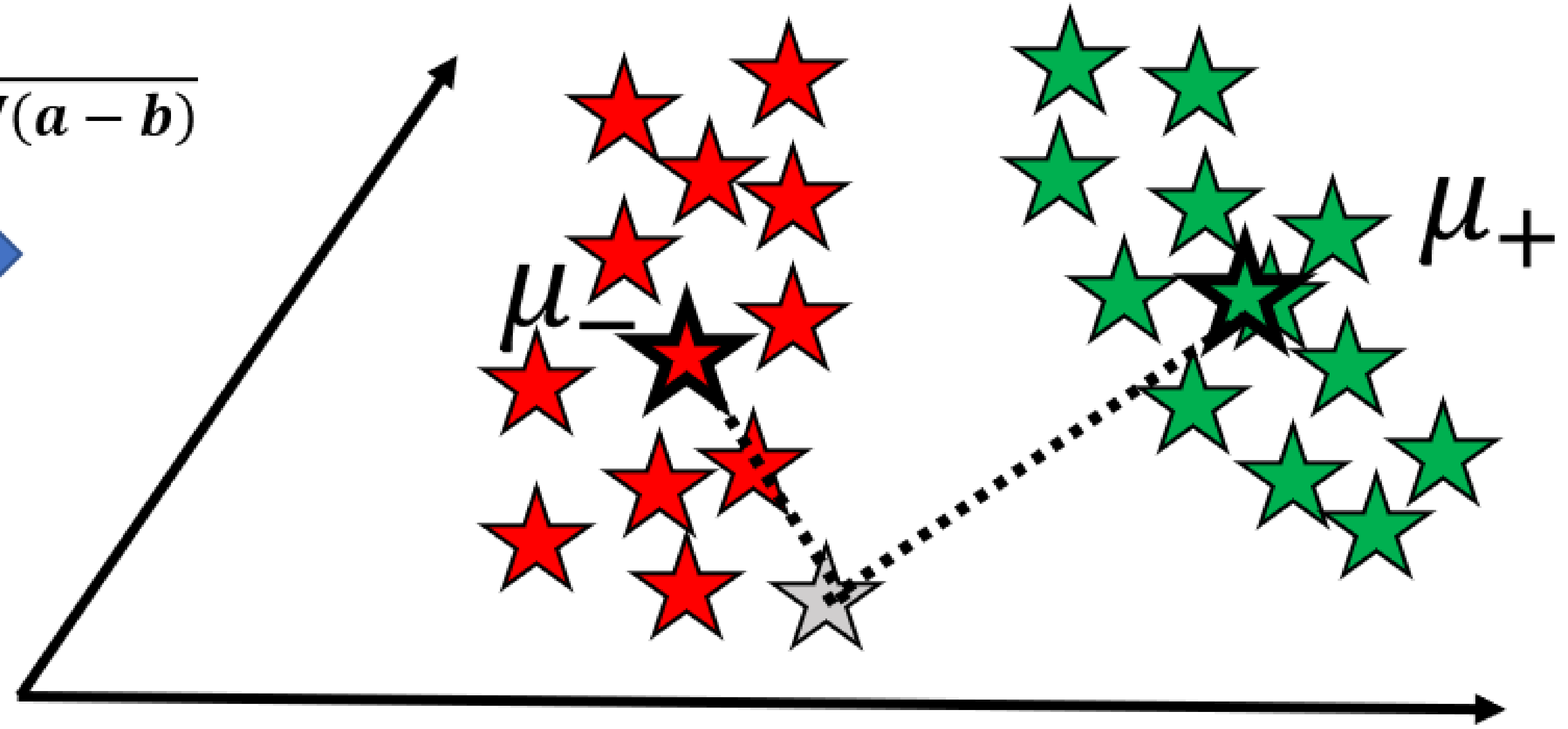
Impact of weighted symmetric matrix

- Mahalanobis distance weights & rotates the axis

W is inverse of 2x2 covariance matrix

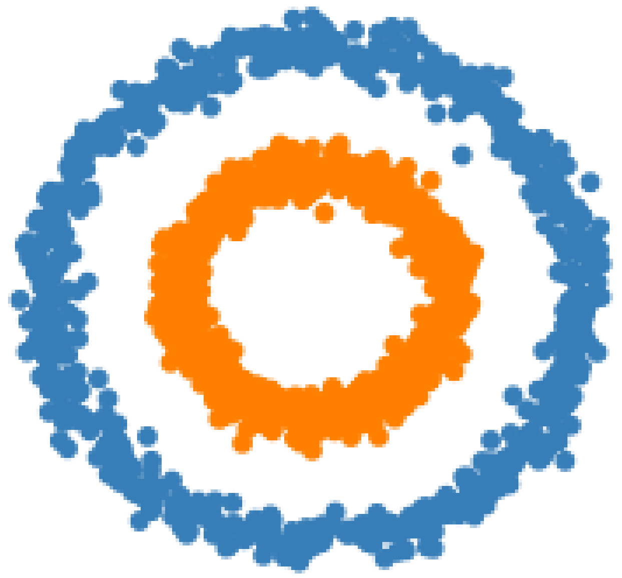


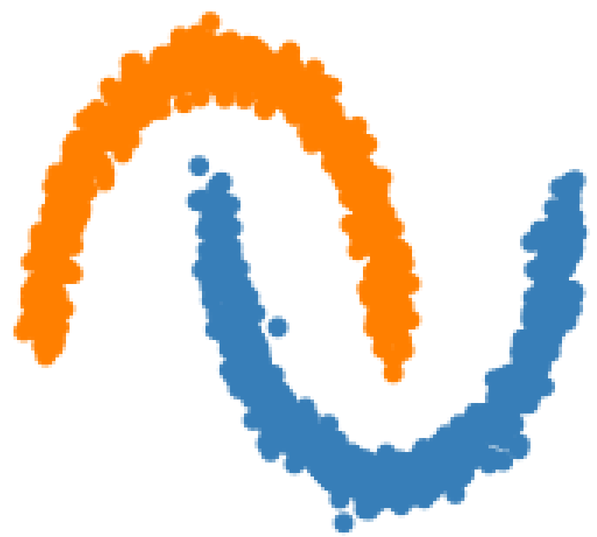
$$d_w(\boldsymbol{a},\boldsymbol{b}) = \sqrt{(\boldsymbol{a}-\boldsymbol{b})^\top \mathbf{W}(\boldsymbol{a}-\boldsymbol{b})}$$



Nearest Centroid – Failure cases

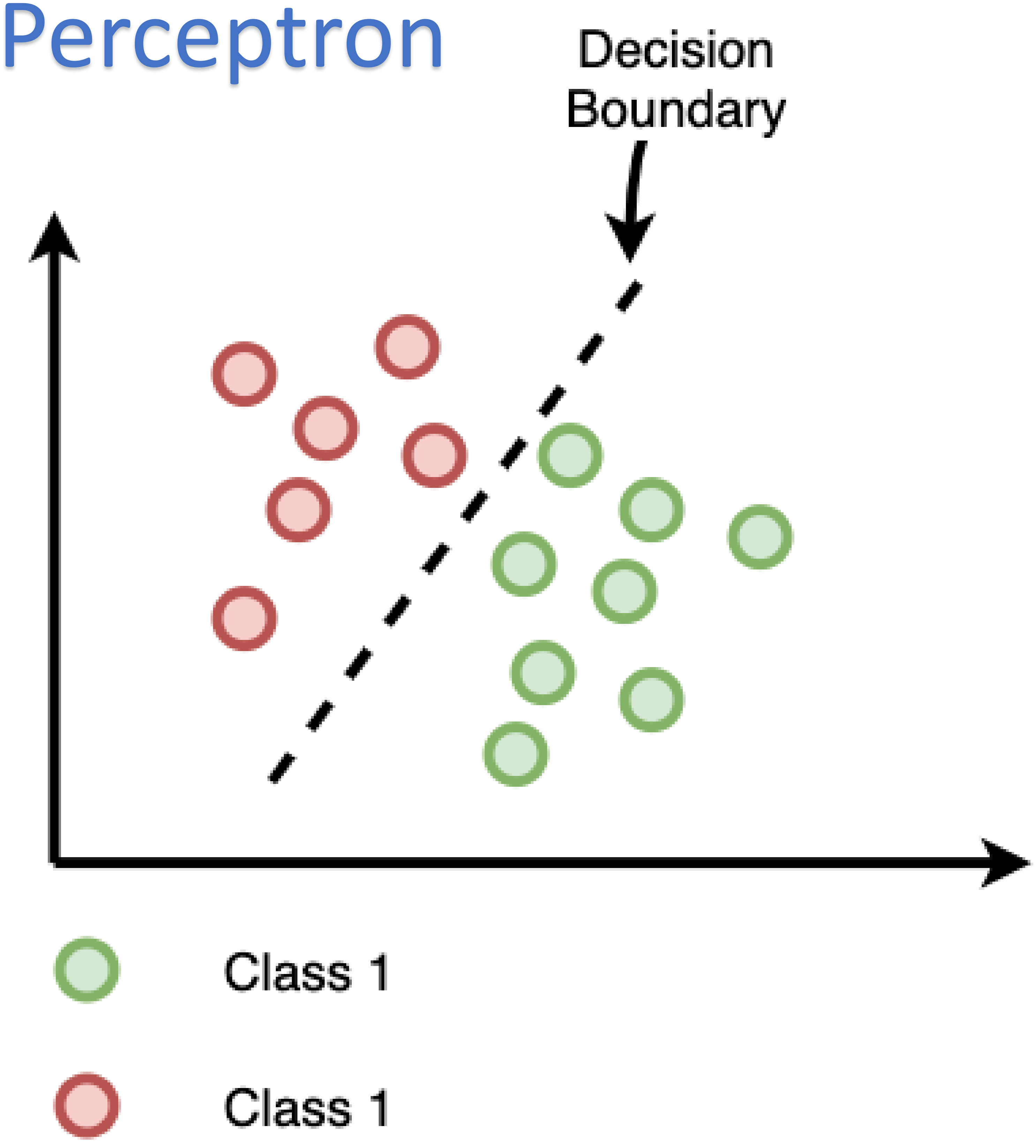
- Learning non linear distance metric using kernels (shortly)
- Learning distance metric with neural networks



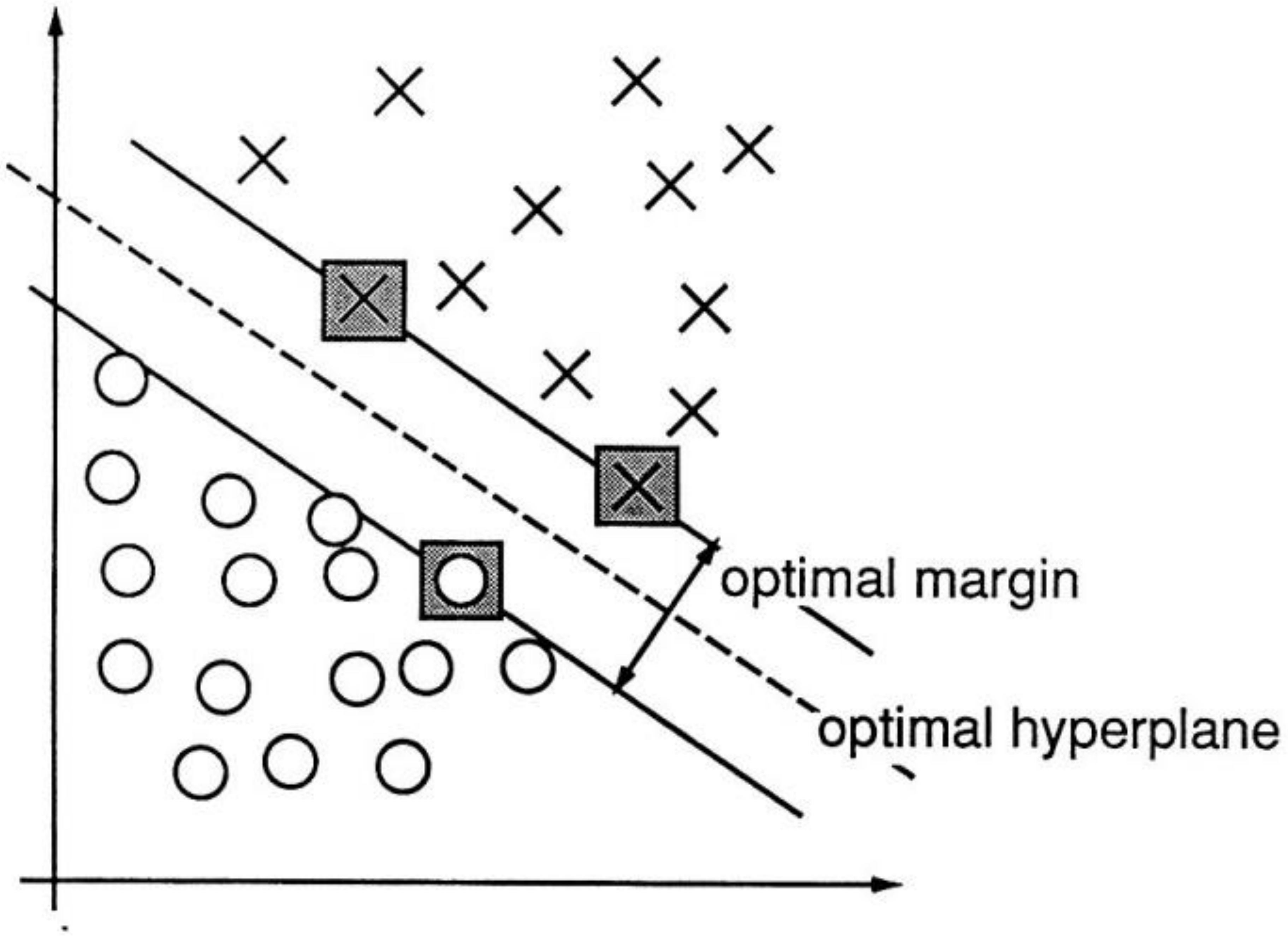


Linear decision boundary

Perceptron

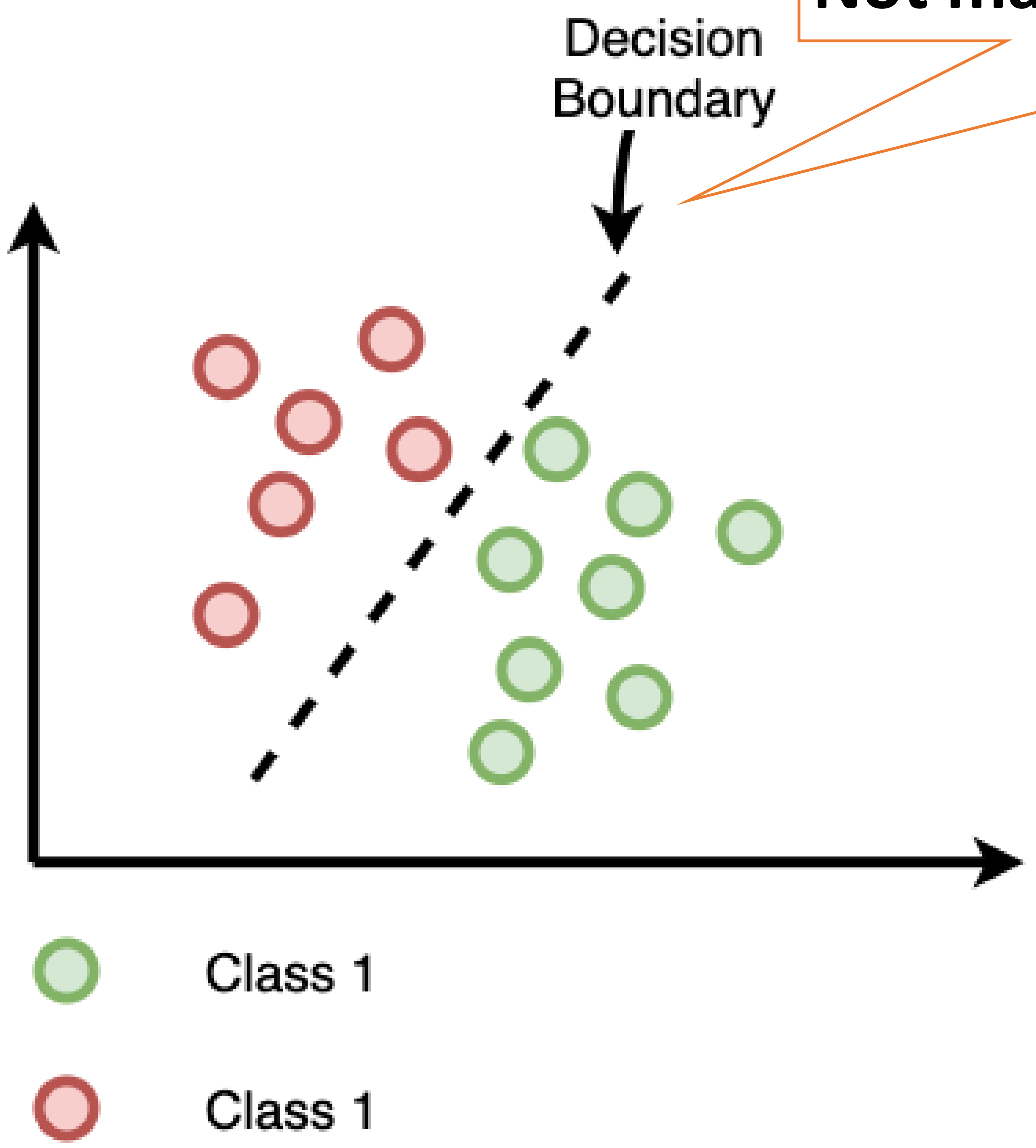


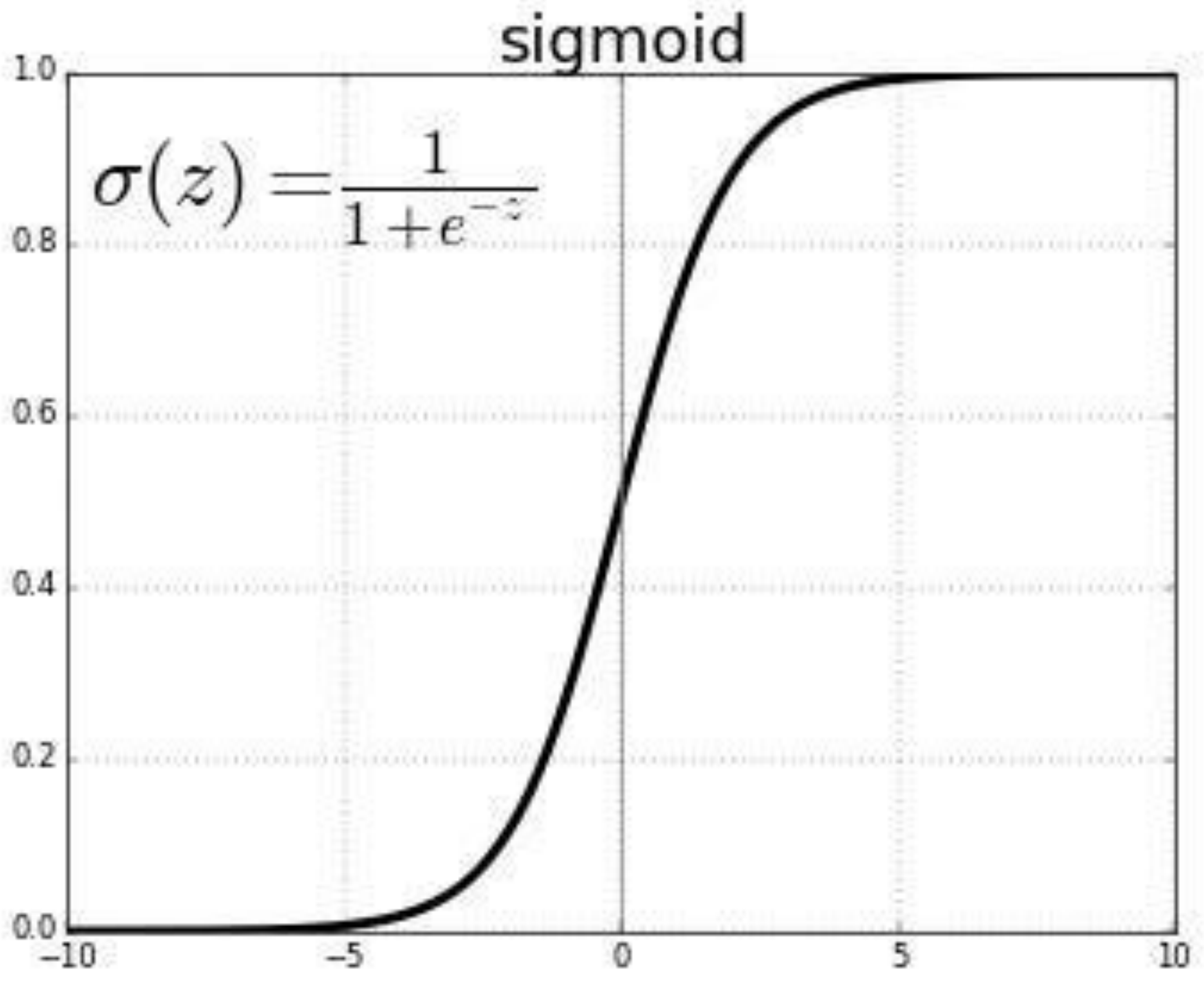
Support Vector Machine



Logistic Regression

Only a conceptual decision boundary. Not mathematically modelled like that



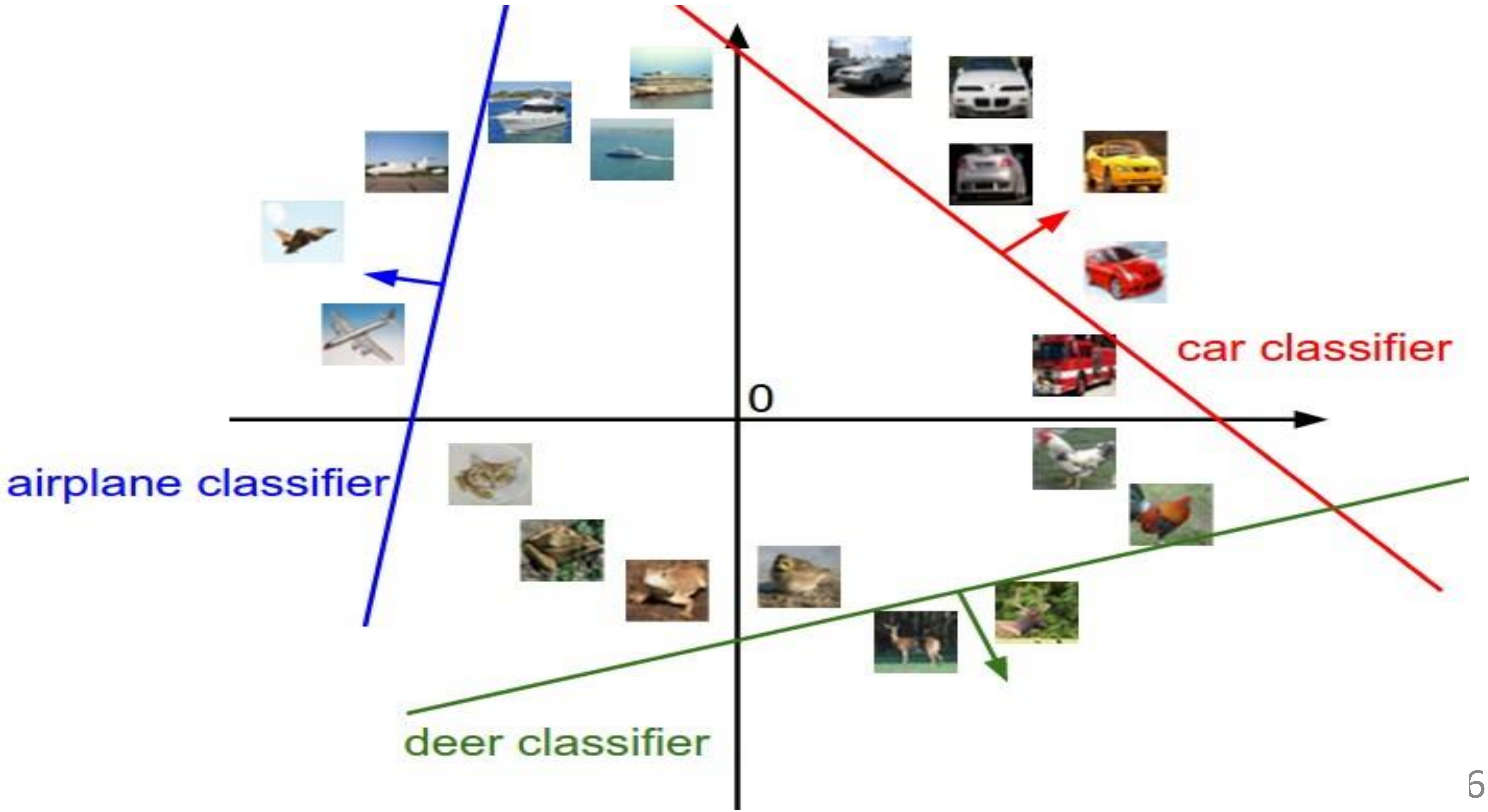


Logistic Regression is Classification

Logistic Regression naam sunke Regression samjha kya

Regression nahin classification hai

One v/s all decision boundary

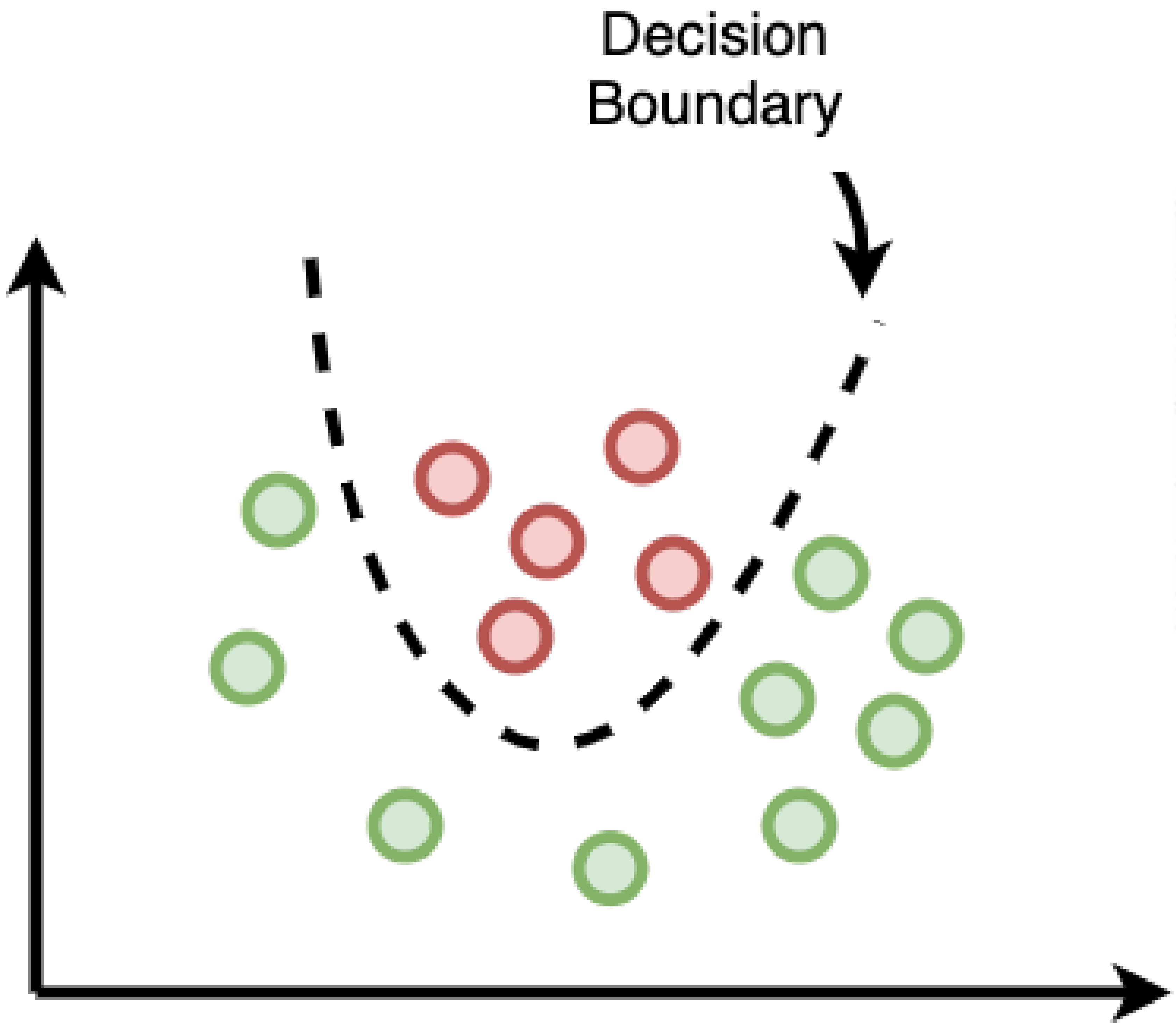


BUT BUT BUT….

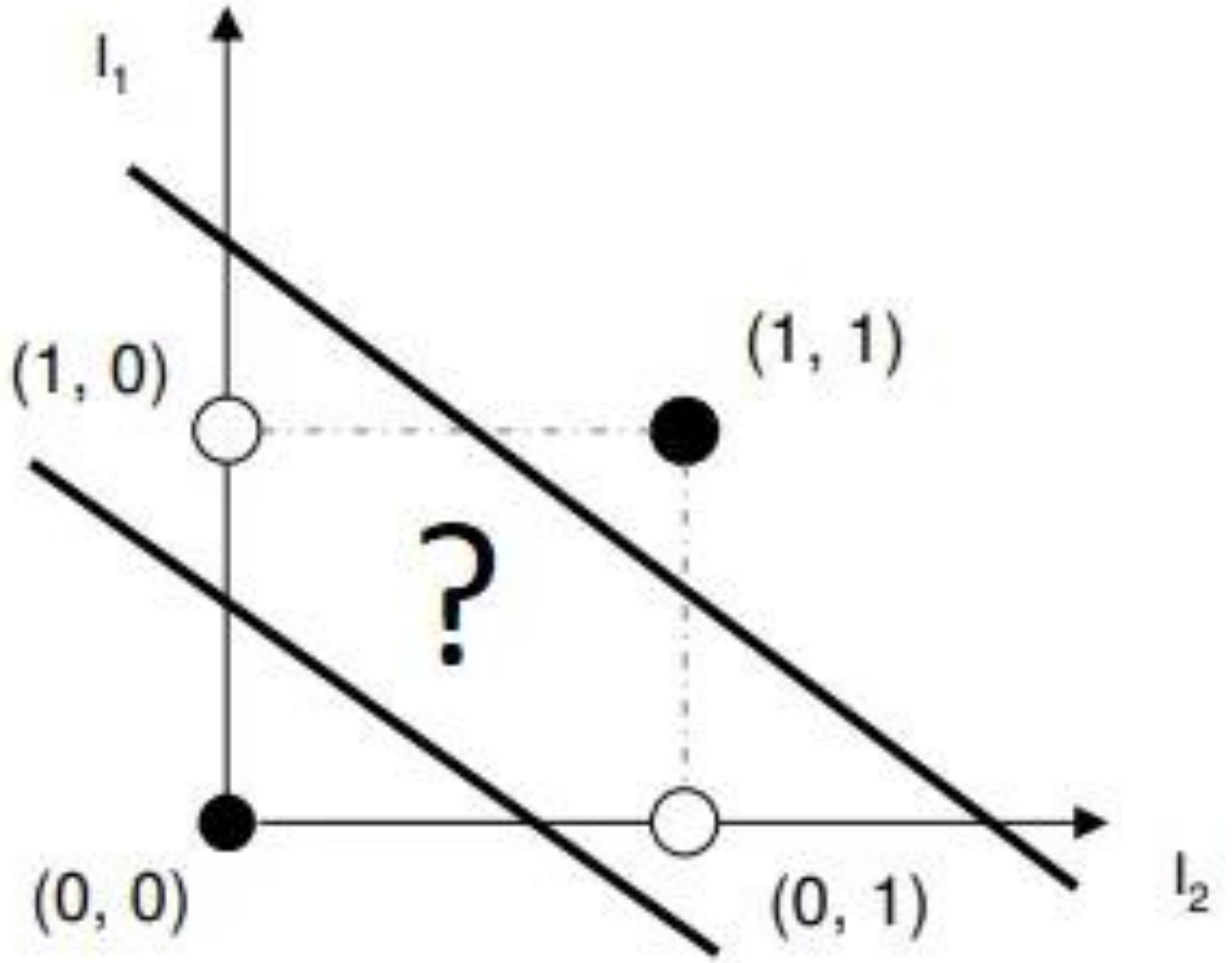
- Real world data almost never linearly separable
- Shouldn’t ML focus on non-linear separability?
  - Yes and No

Non linear decision boundary

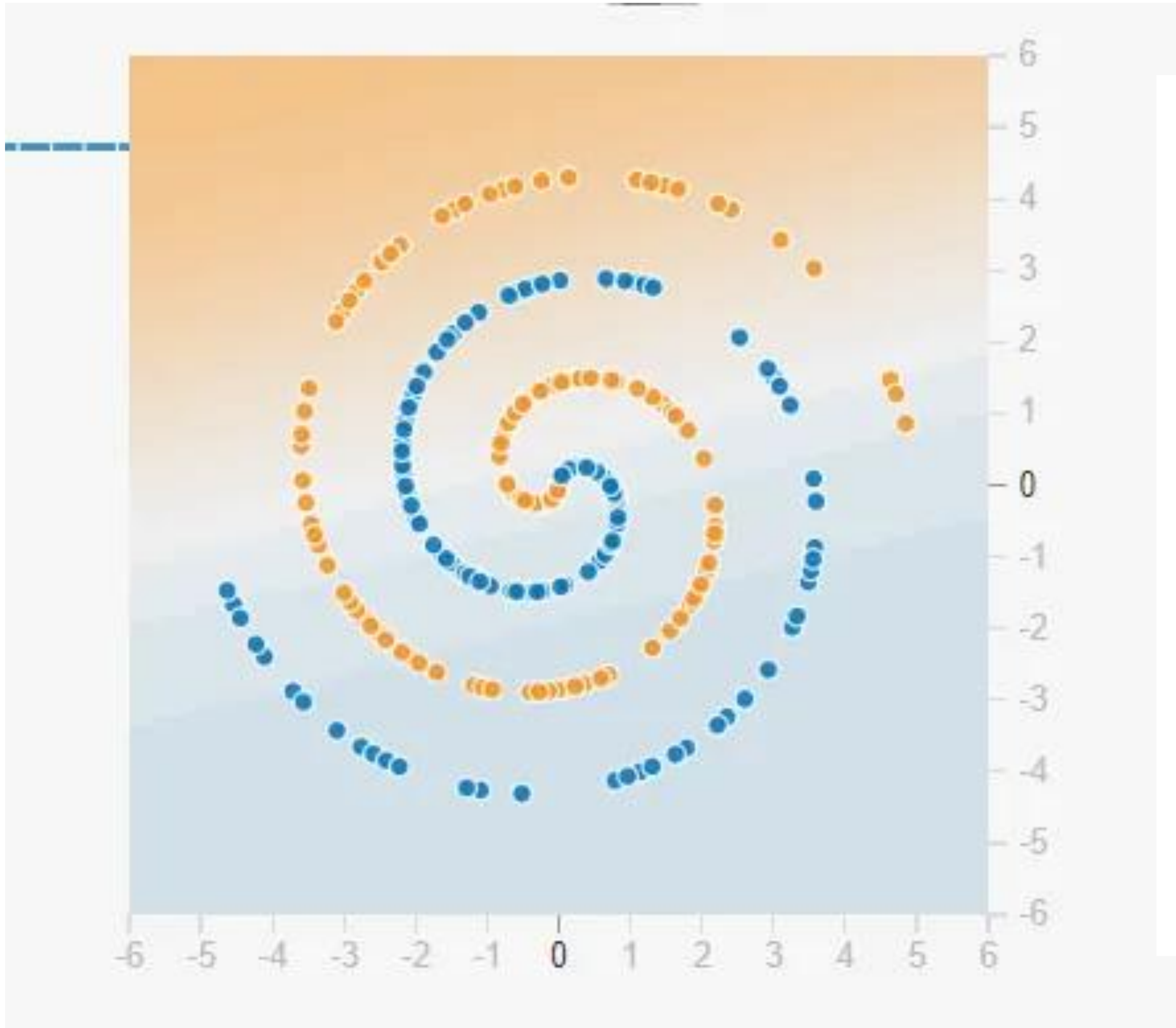
- Directly create a non linear decision boundary

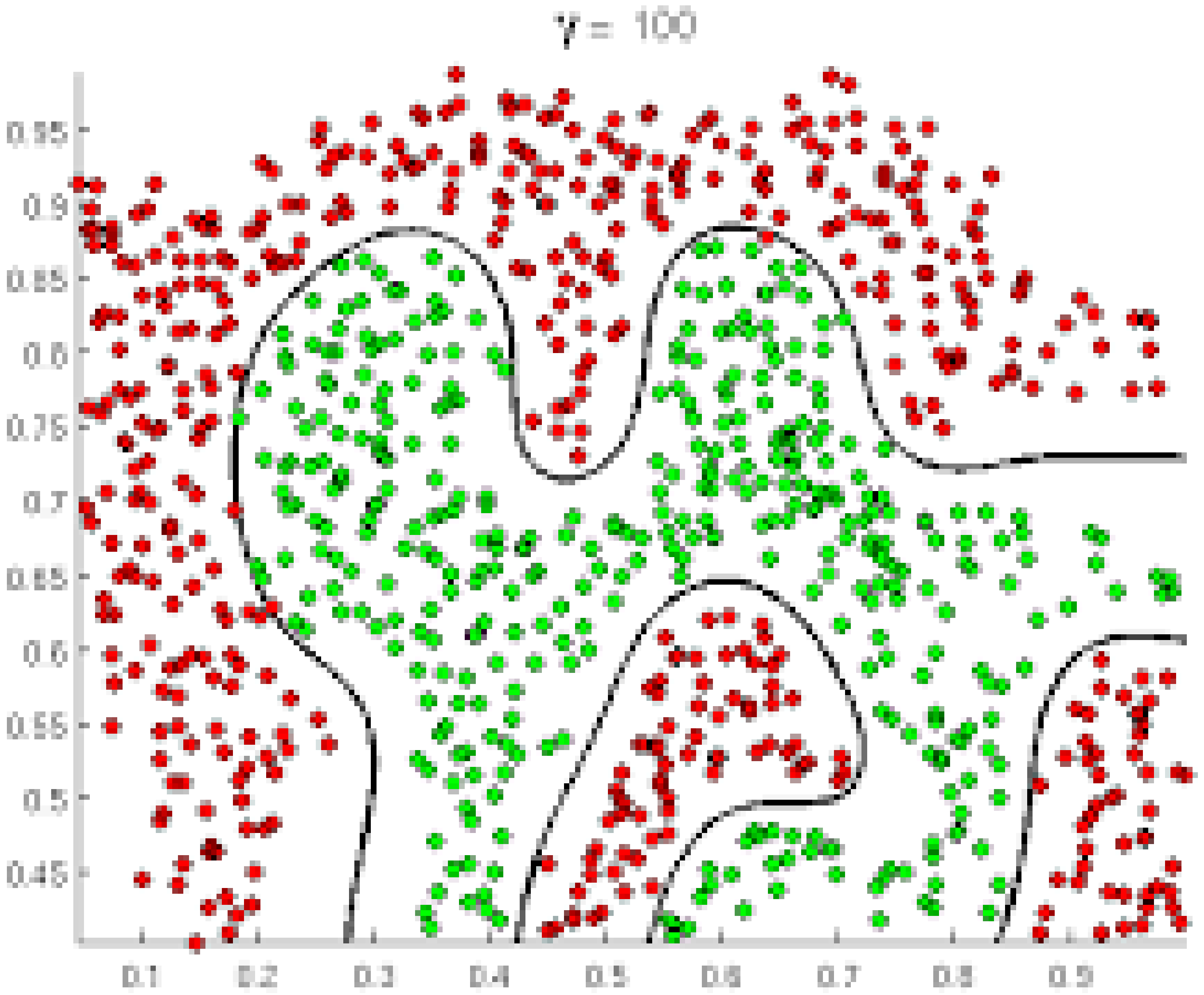


XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Non linear decision boundary



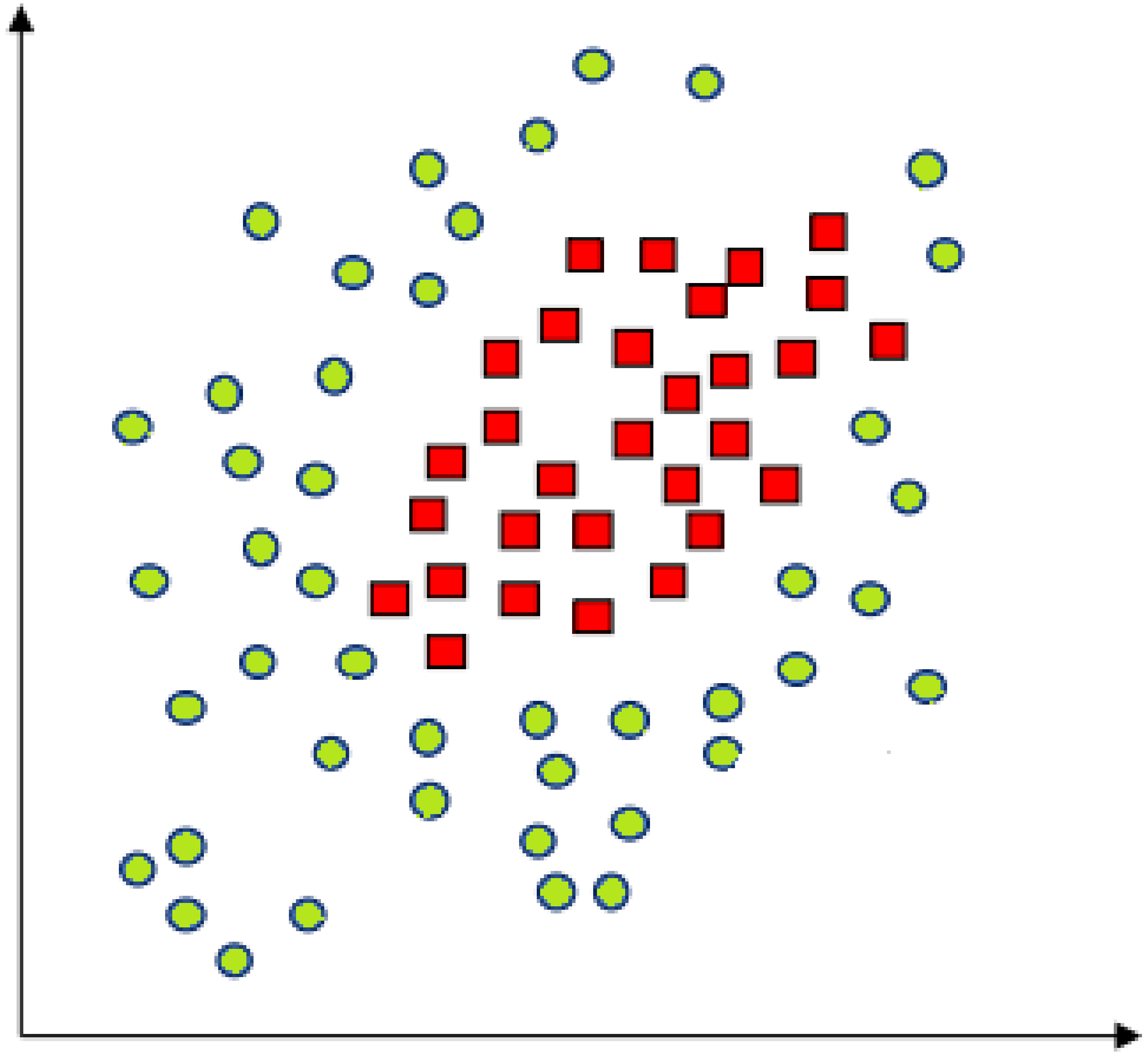


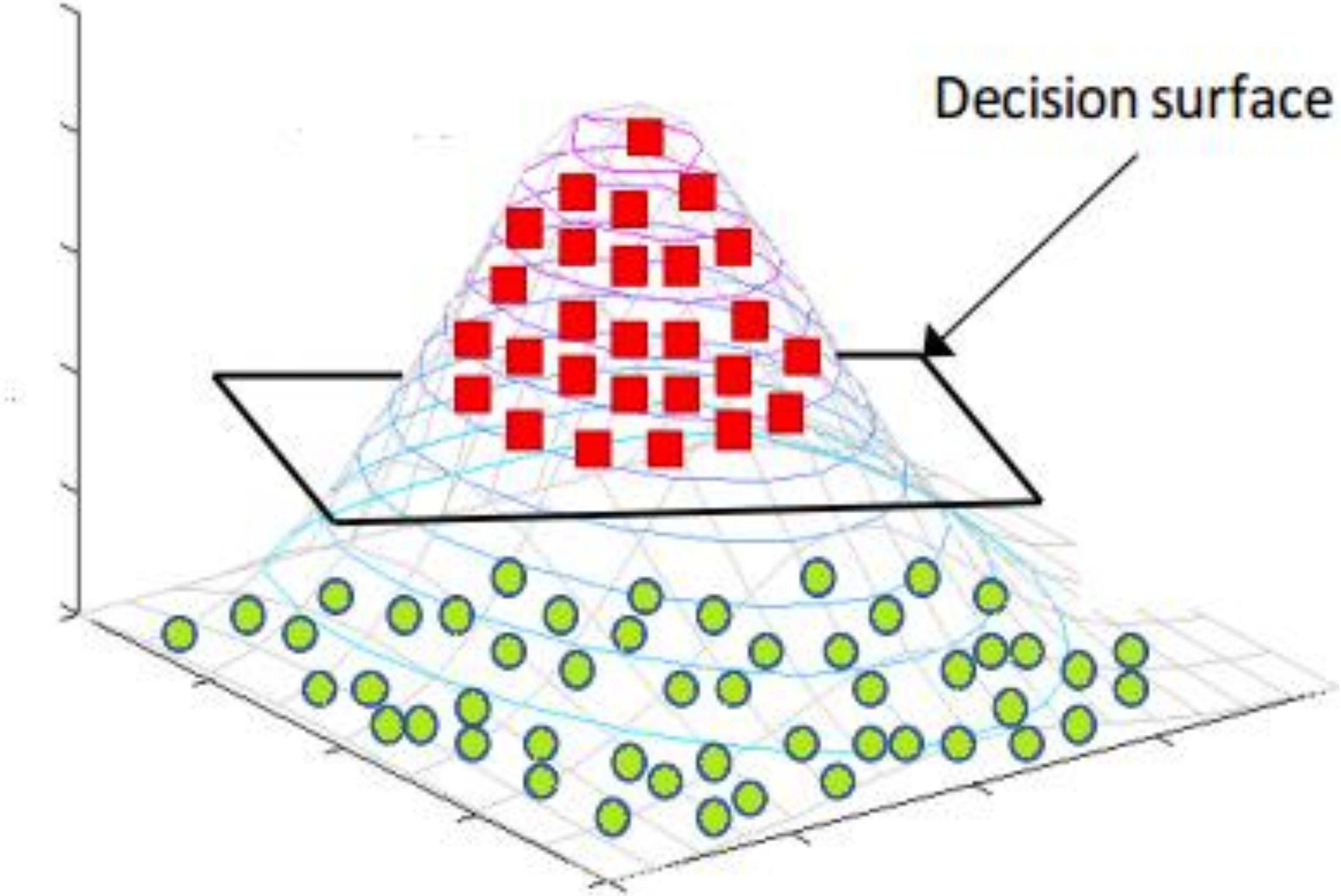
The hard problem

- Solving in non linear domain is hard
- That’s why we do complex Feature Transformation
  - Not just simple transformation covered earlier
  - Kernel transformation
  - Complex distance metric
    - Involves stretching, squishing & folding the space
- Finally solve the problem in linear domain ☺

Non linear decision boundary

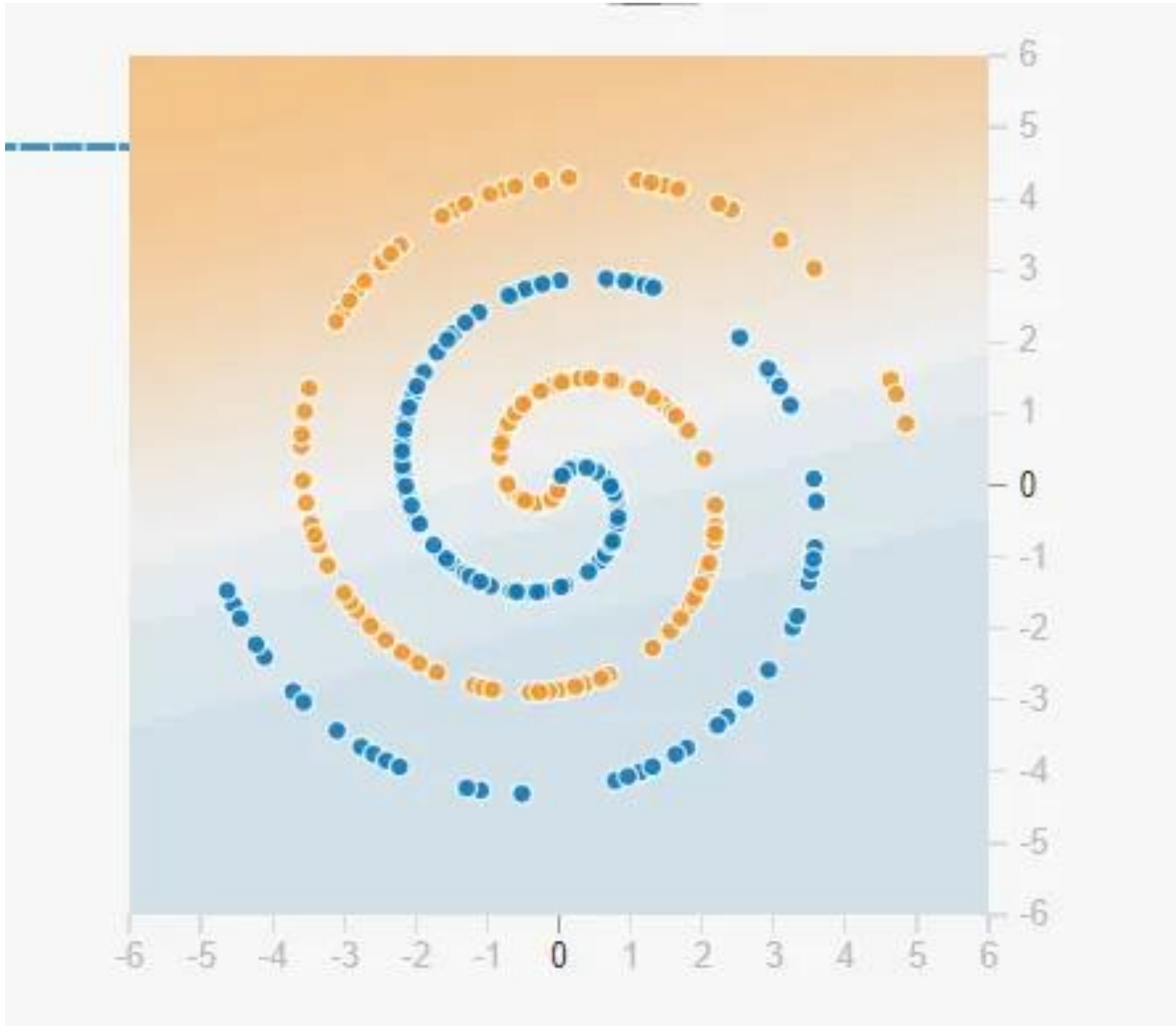
- https://www.geogebra.org/calculator/ve2earrn
- https://www.geogebra.org/m/pO4JcWPz

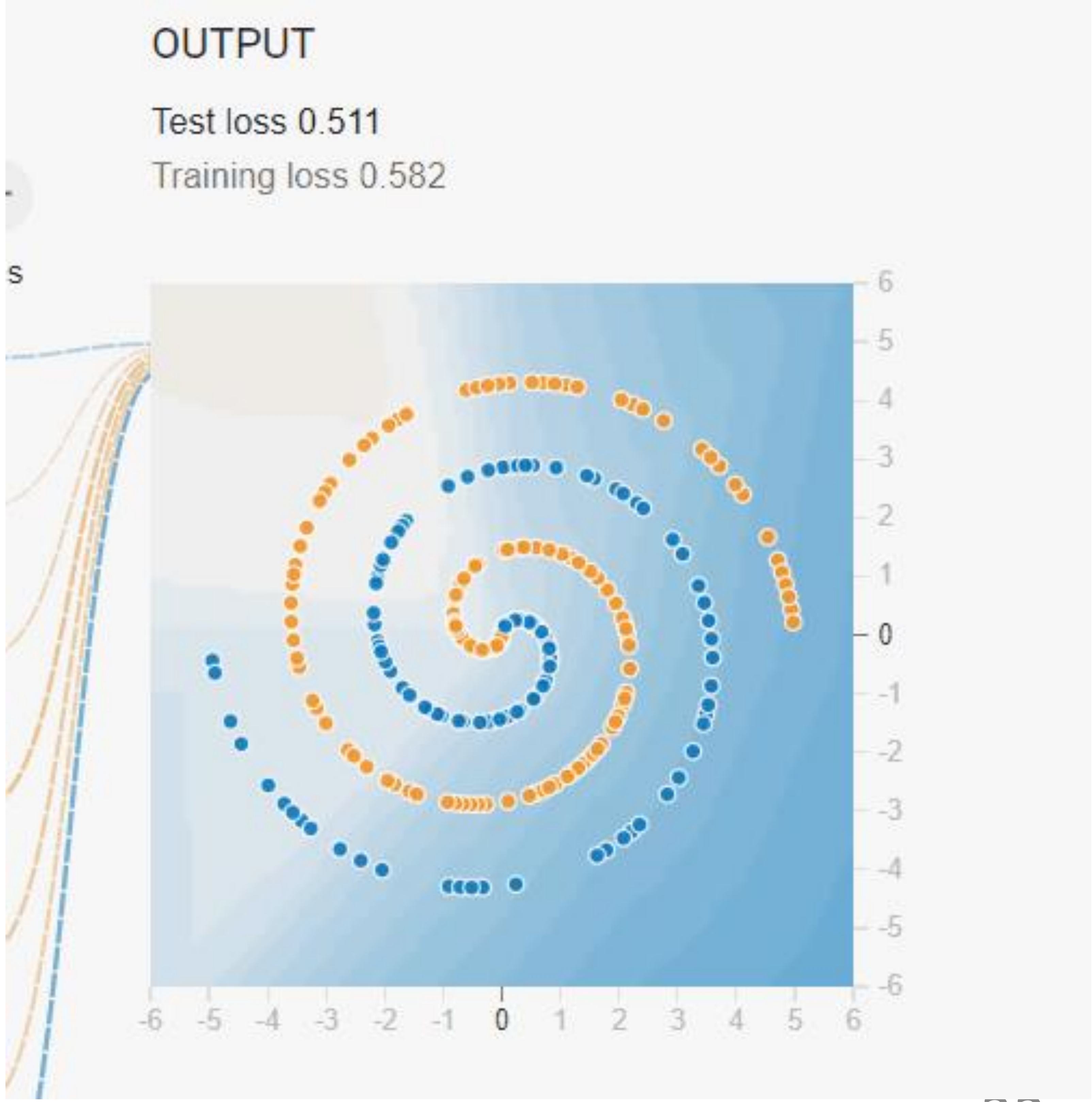




Arbitrary transformations

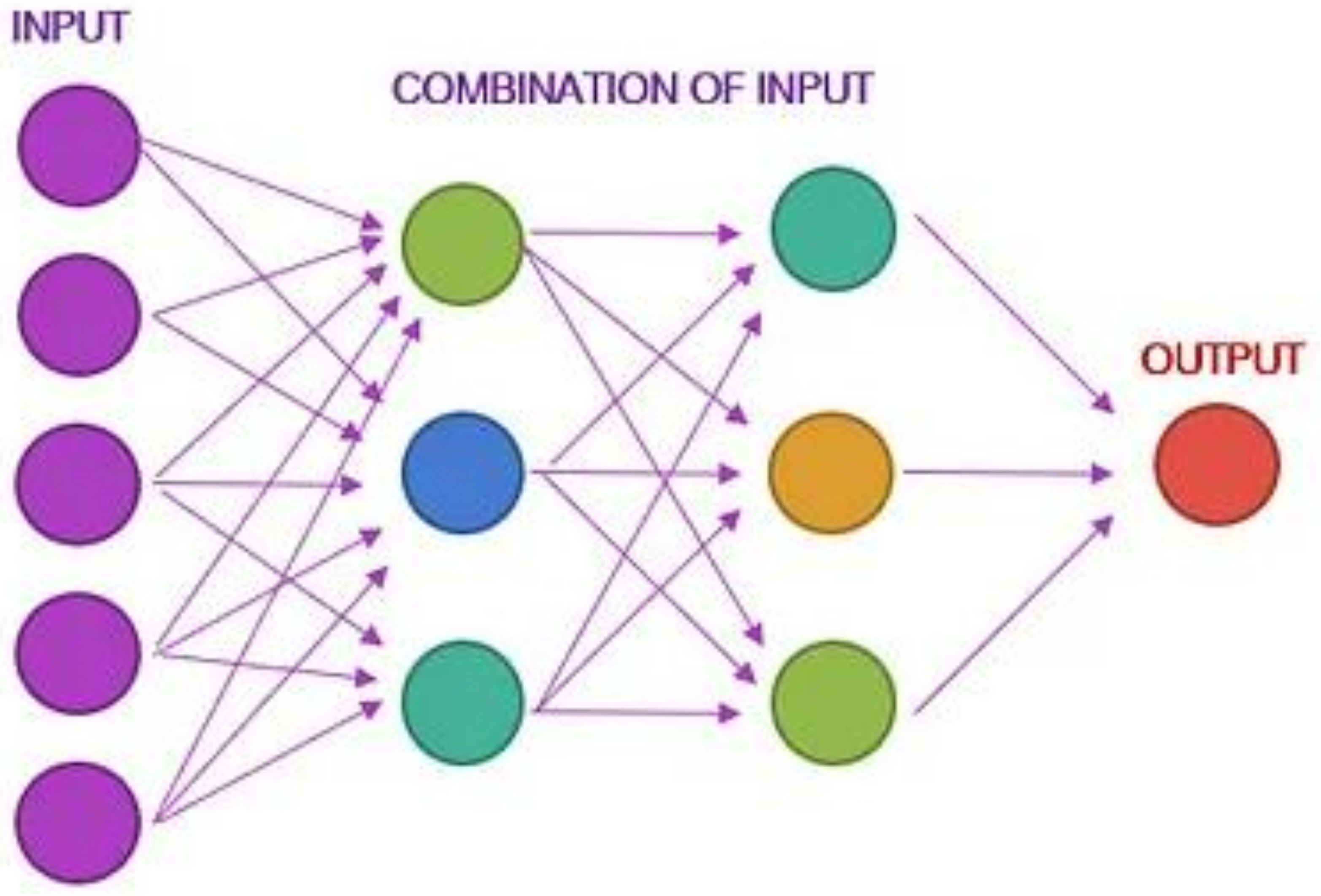
- With neural network

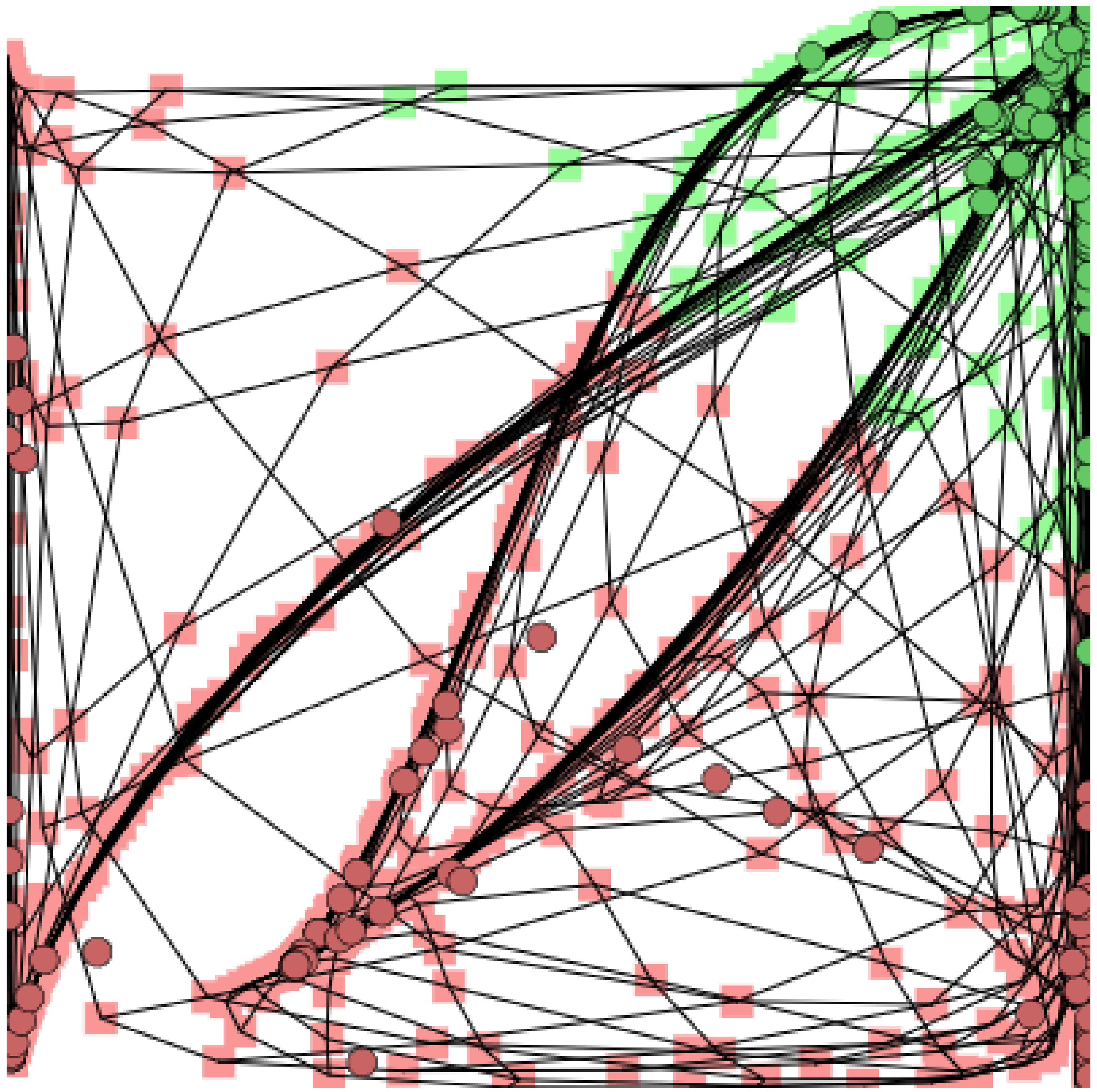




Arbitrary transformations

- By folding the space to apply linear classifier
- https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html



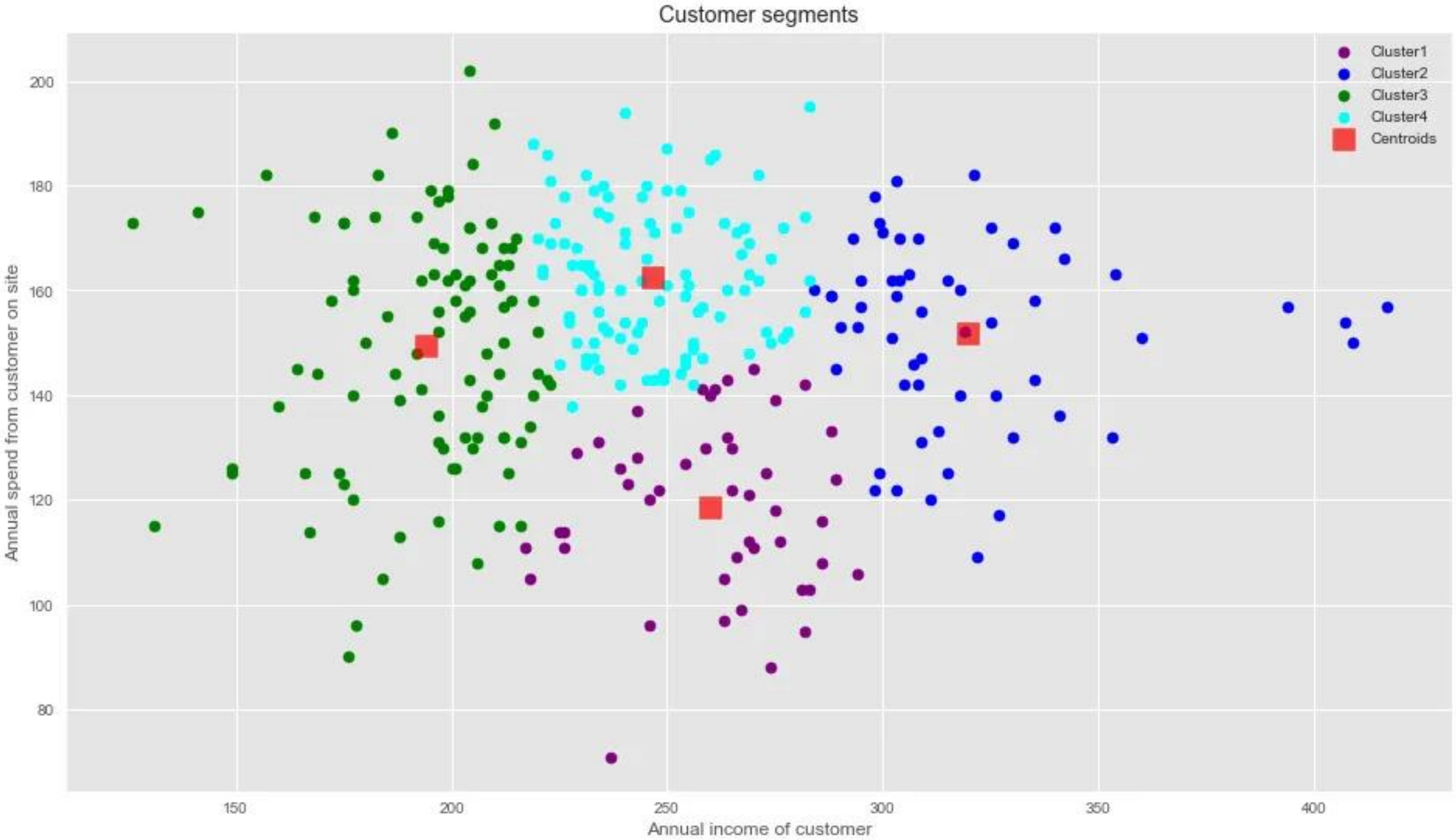


Optional reading

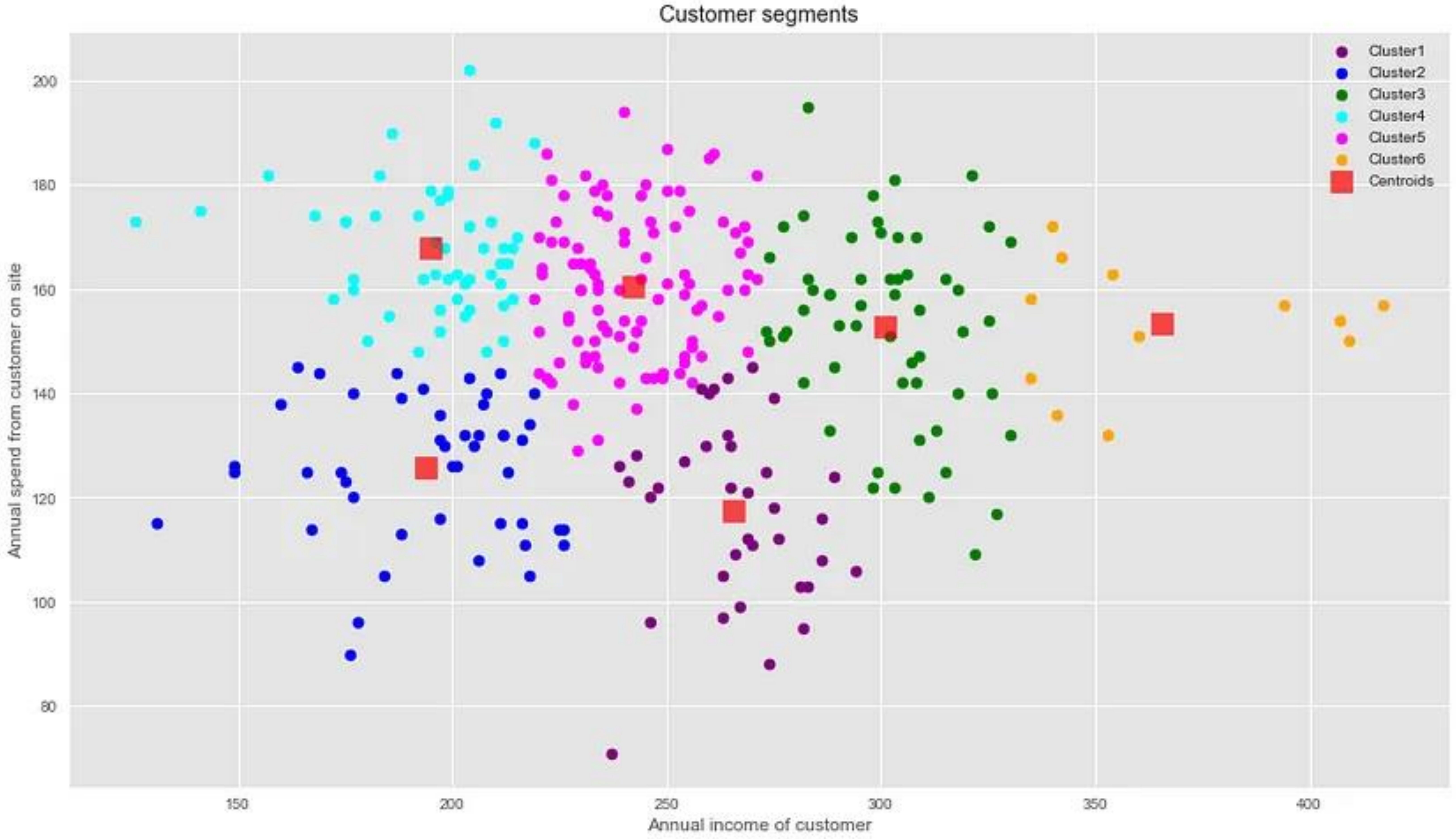
- Neural Networks Manifold and Topology
- A tutorial on distance metric learning

Unsupervised Learning

Clustering

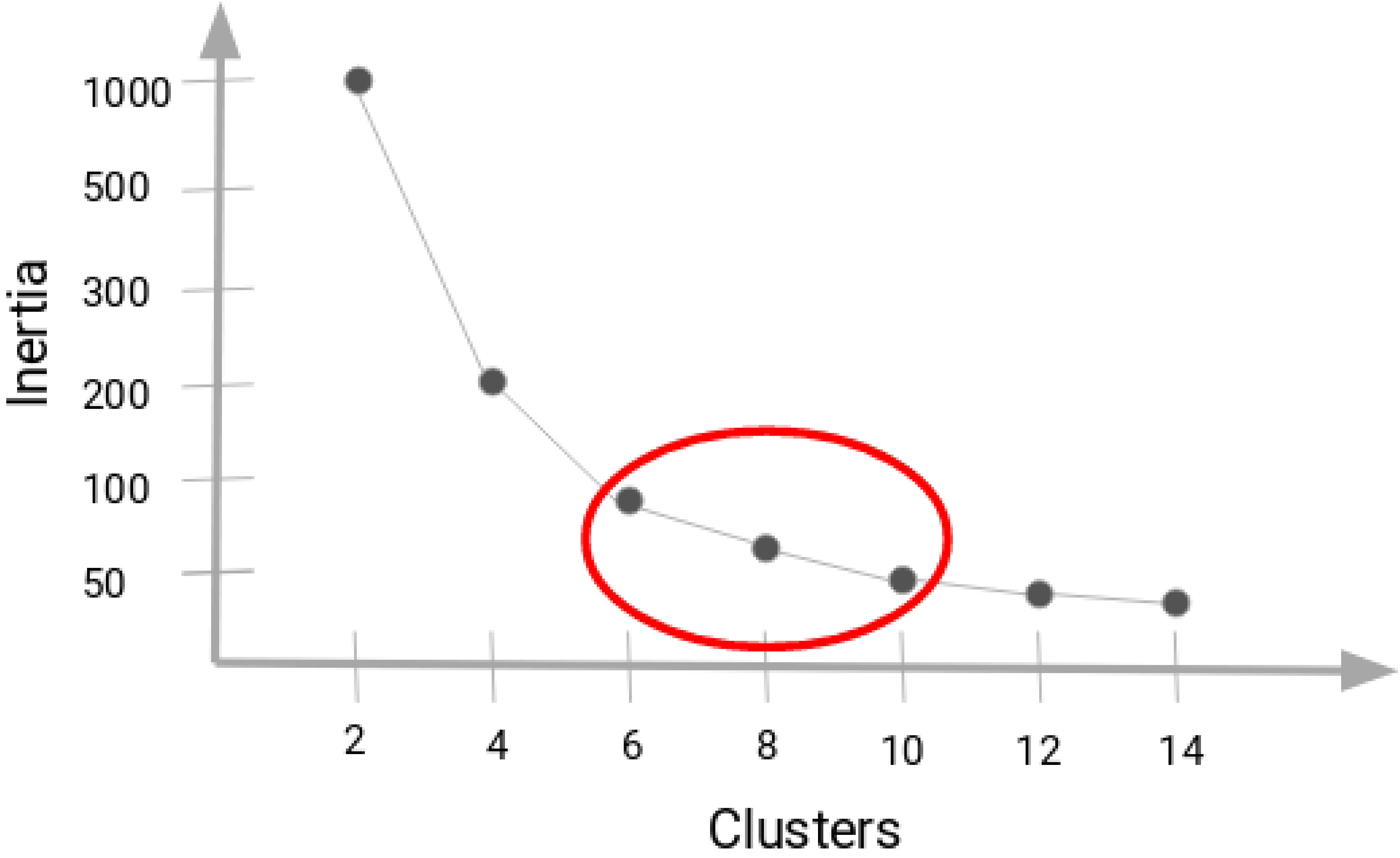

Customer segments
Cluster1
Cluster2
Cluster3
Cluster4
Centroids
200
180
160
140
120
100
80
150
200
250
300
350
400
Annual spend from customer on site
Annual income of customer

Clustering



Customer segments
Cluster1
Cluster2
Cluster3
Cluster4
Cluster5
Cluster6
Centroids
200
180
160
140
120
100
80
Annual spend from customer on site
150
200
250
300
350
400
Annual income of customer

Clustering




QUESTIONS

Literature Survey