Machine Learning – How it really works

Souvik Chanda

Agenda

- Why are we having this session?
- Machine learning – what, why & where?
- Types of Machine Learning
- How Supervised Learning works – Illustration using Regression
- Classification – Basic Understanding
- Unsupervised learning
- Machine Learning Development Lifecycle
- Fine Tuning your model
- Bias & variance
- Machine Learning Development Life Cycle
- Take Away

Why are we having this session?

Usual state of understanding

- Very high level idea
- Often not clear on what really goes on or how is it different from traditional programming logic
- Lack of working knowledge

Desired state of understanding

- To clear the myth
- To understand how it really works
- Apply the understanding in work

Problem

Machine learning really is maths

- Linear algebra
- Calculus
- Statistics & probability

Promise

I **solemnly** promise to keep the maths to the minimum!

Cognizant

Machine learning – what, why & where?

Informal Definition

Field of study that gives computers the ability to learn without being explicitly programmed.
Arthur Samuel (1959).

Formal Definition

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.
Tom Mitchell (1998)

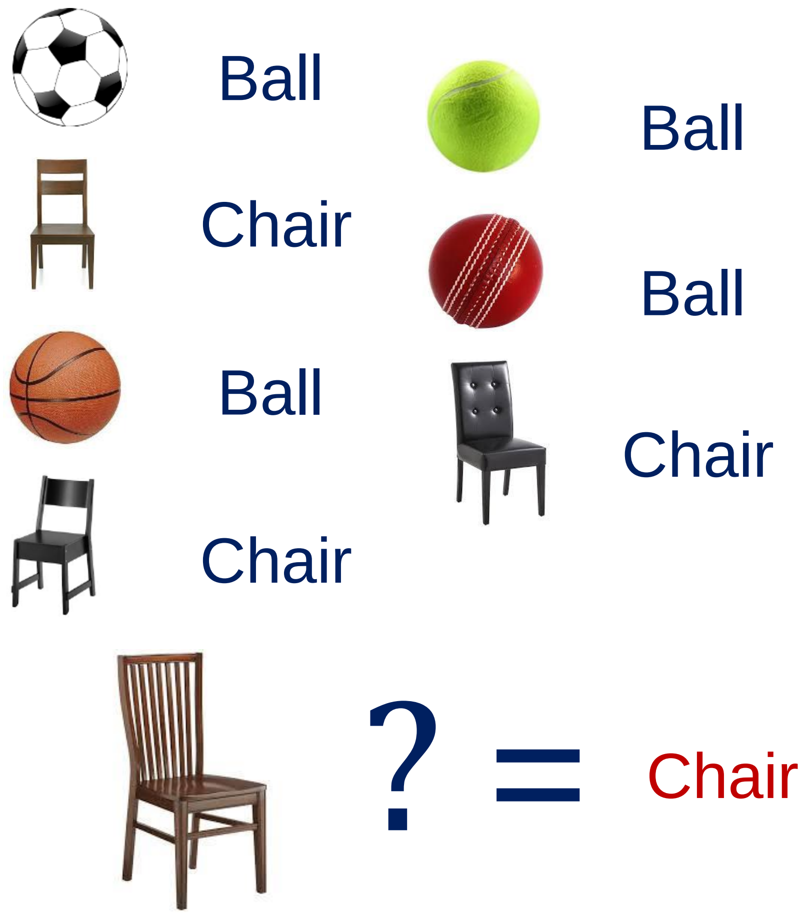
Applications in problems where it's near impossible to program all if-then-else conditions

- Game of chess
- Natural language processing
- Driving a car
- Recommending the next movie to watch
- Identifying cancer
- Recognizing faces & voices
- Churn prediction, Automated underwriting & claims handling

Cognizant

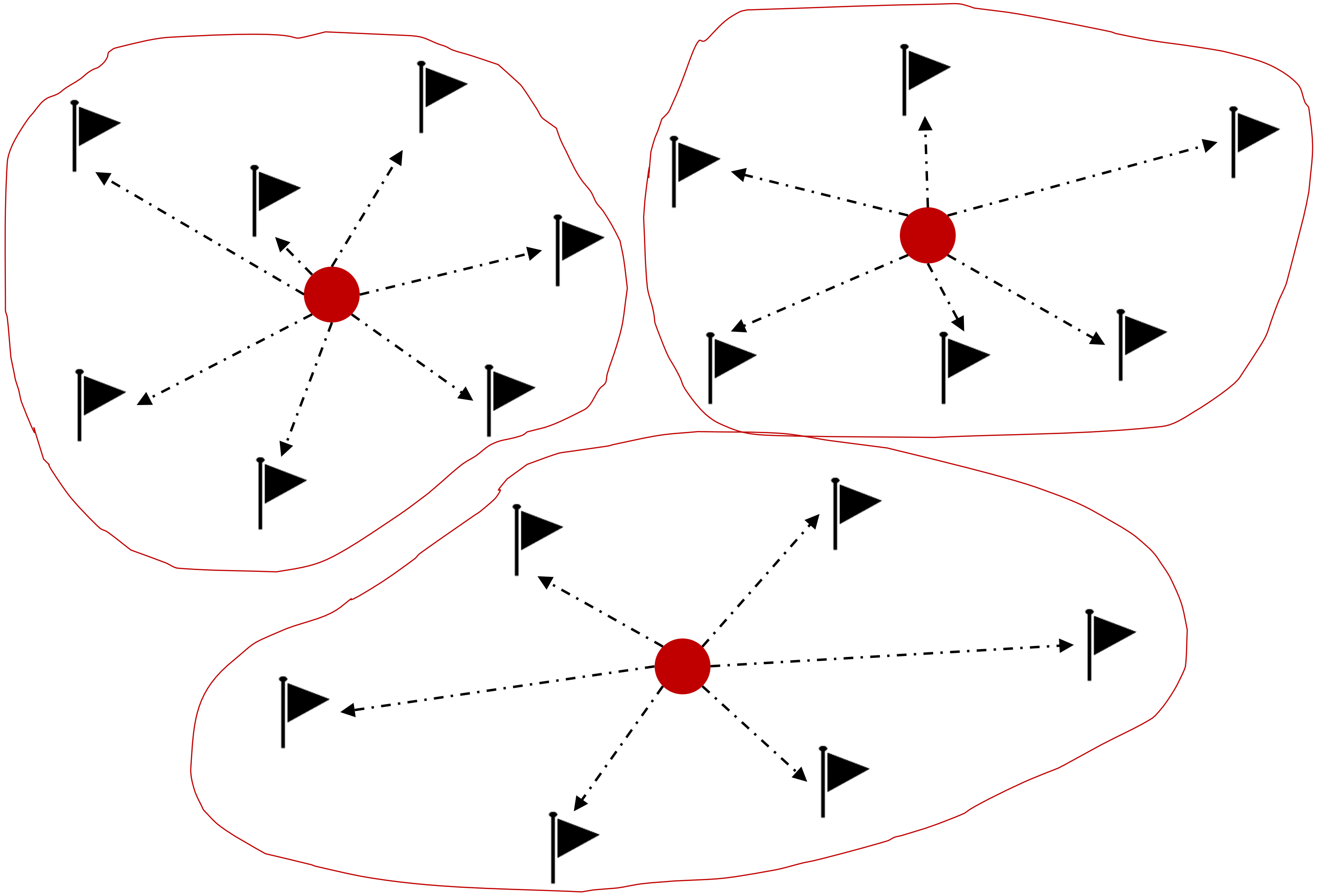
Types of Machine Learning – Supervised vs Unsupervised

Supervised Learning uses labelled data



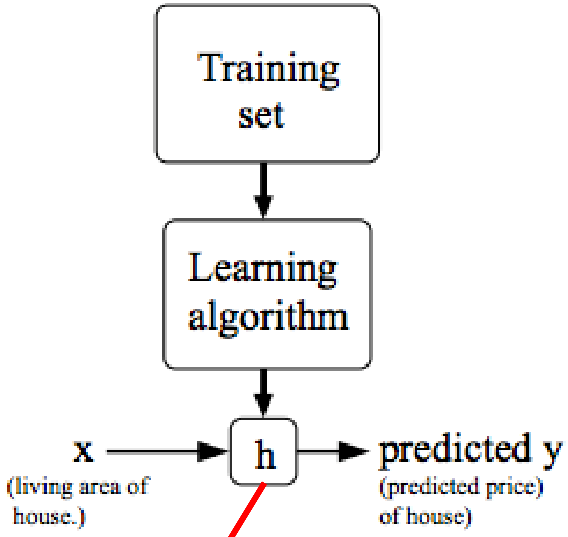
Example: Classification

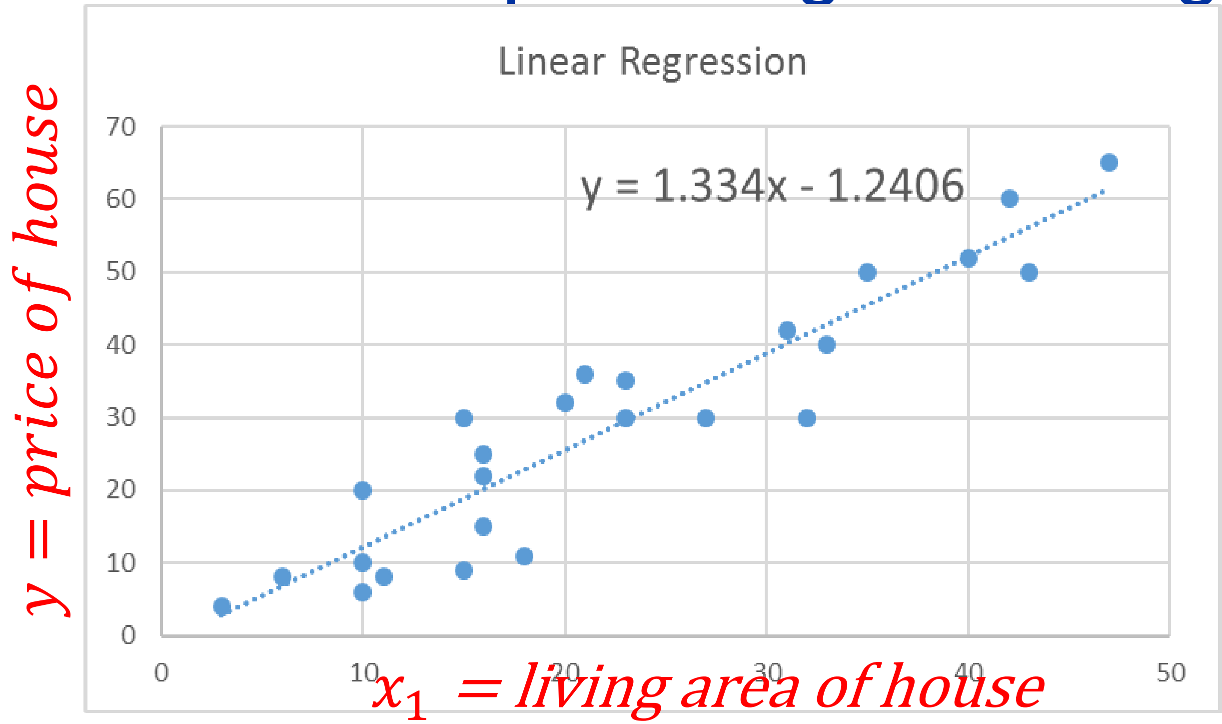
Unsupervised Learning uses unlabelled data



Example: Clustering

How does it really work – An example using linear regression





$y = b + mx$

m = slope &

b = y intercept

Equation of a straight line

Generalization of

$y = b + mx$

$\theta_0 + \theta_1 x_1$

Linear Regression

Hypothesis $y = h_\theta(x_i) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots. + \theta_2 x_2$

Given new x_1, *predict* y

Uses Mean Squared Error to fit the model or learn the parameters ie., the values of θ

Learning the parameters using Gradient Descent to minimize cost function

Cost Function = Mean Squared Error

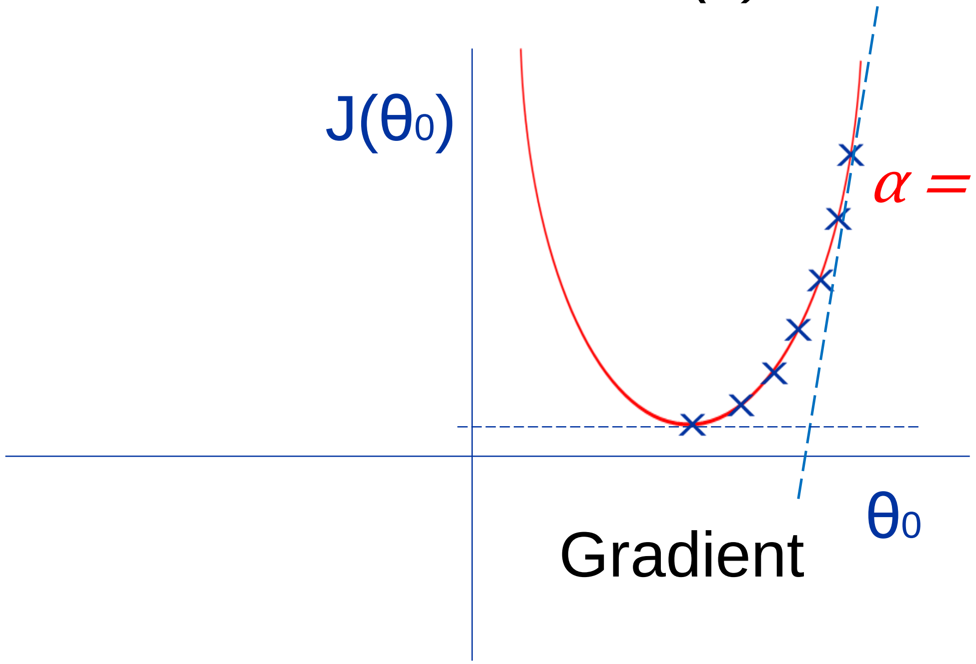
$m = number\ of\ examples$

$\theta\ represents\ the\ parameters\ for\ all\ features$

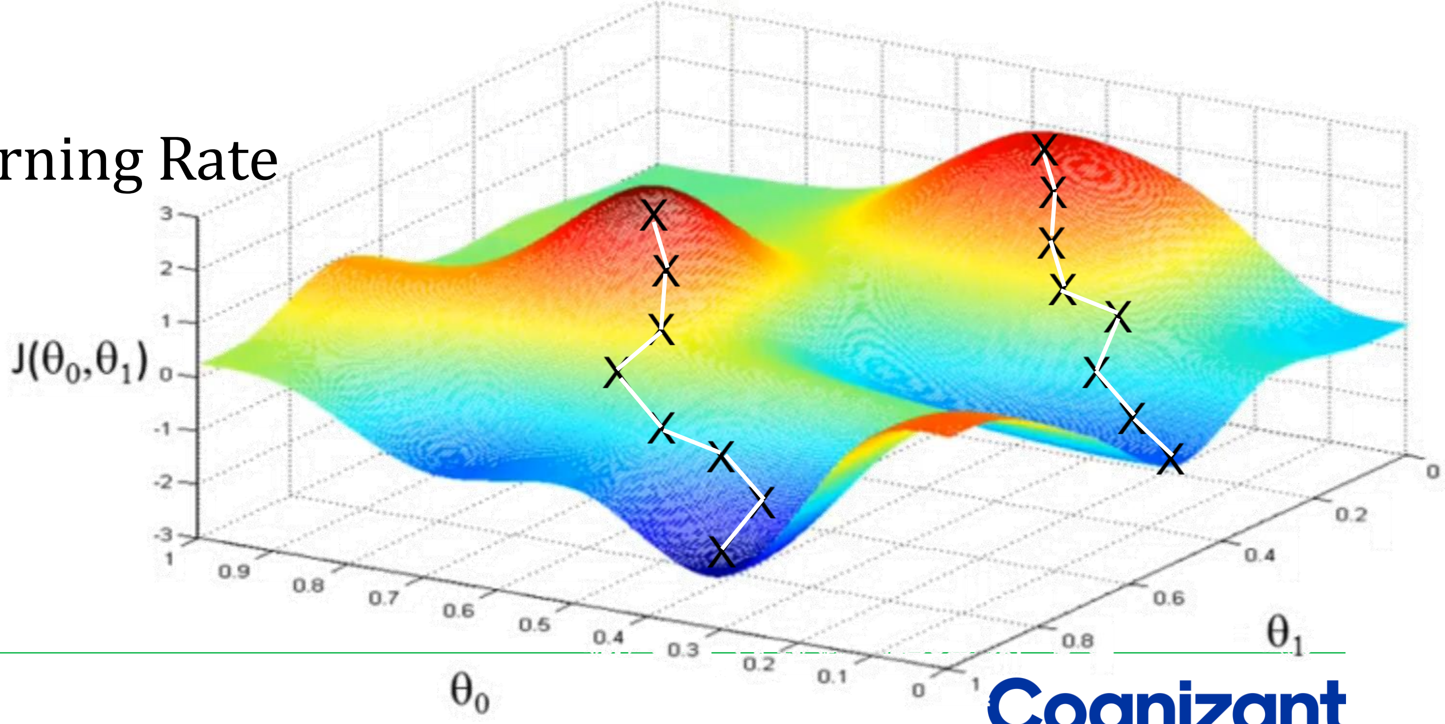
$\lambda = regularization\ parameter$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 + \frac{\lambda}{2m}\sum_{i=1}^{m}\theta_i^2$$

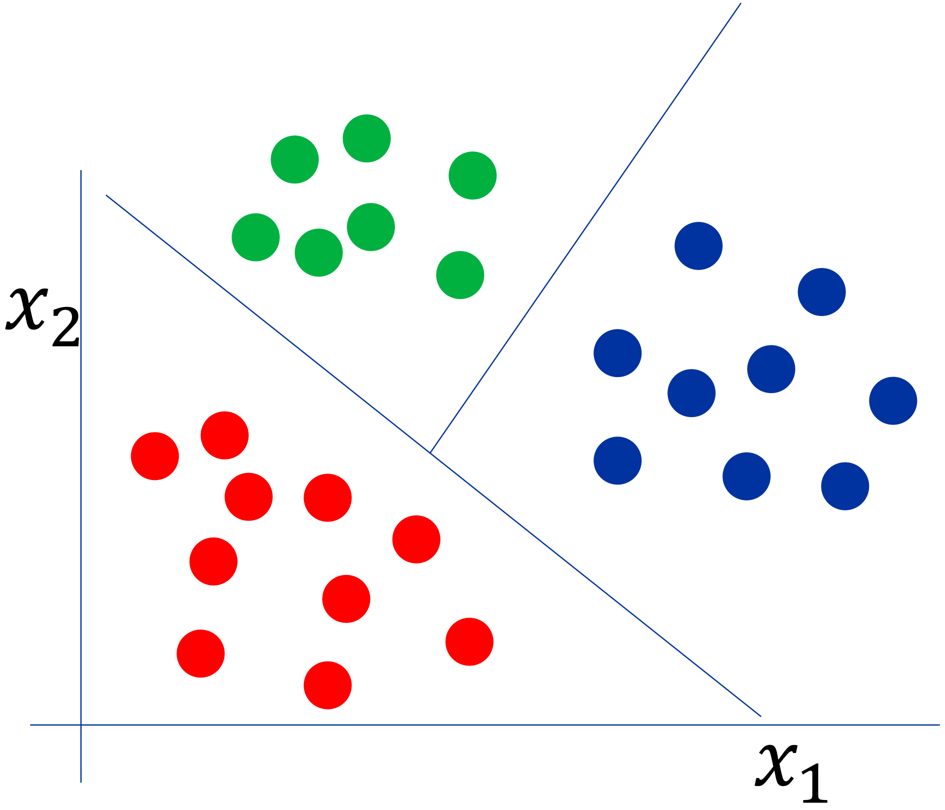
Goal is to minimise J(θ) with respect to the parameters θ1, θ2….



$\alpha =$ Learning Rate

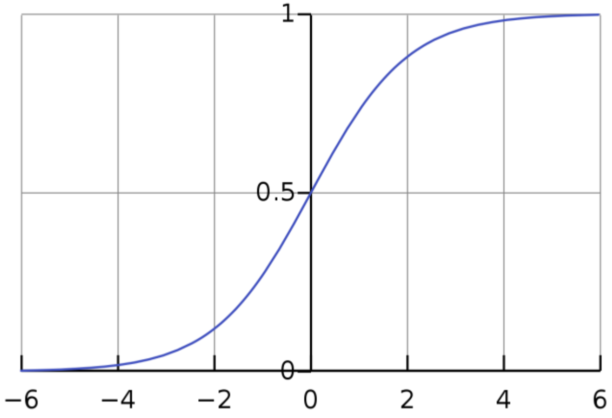


Basics of Classification using Logistic Regression



Binary Classification

$$h_\theta(x) \geq 0.5 \rightarrow y=1$$
$$h_\theta(x) < 0.5 \rightarrow y=0$$



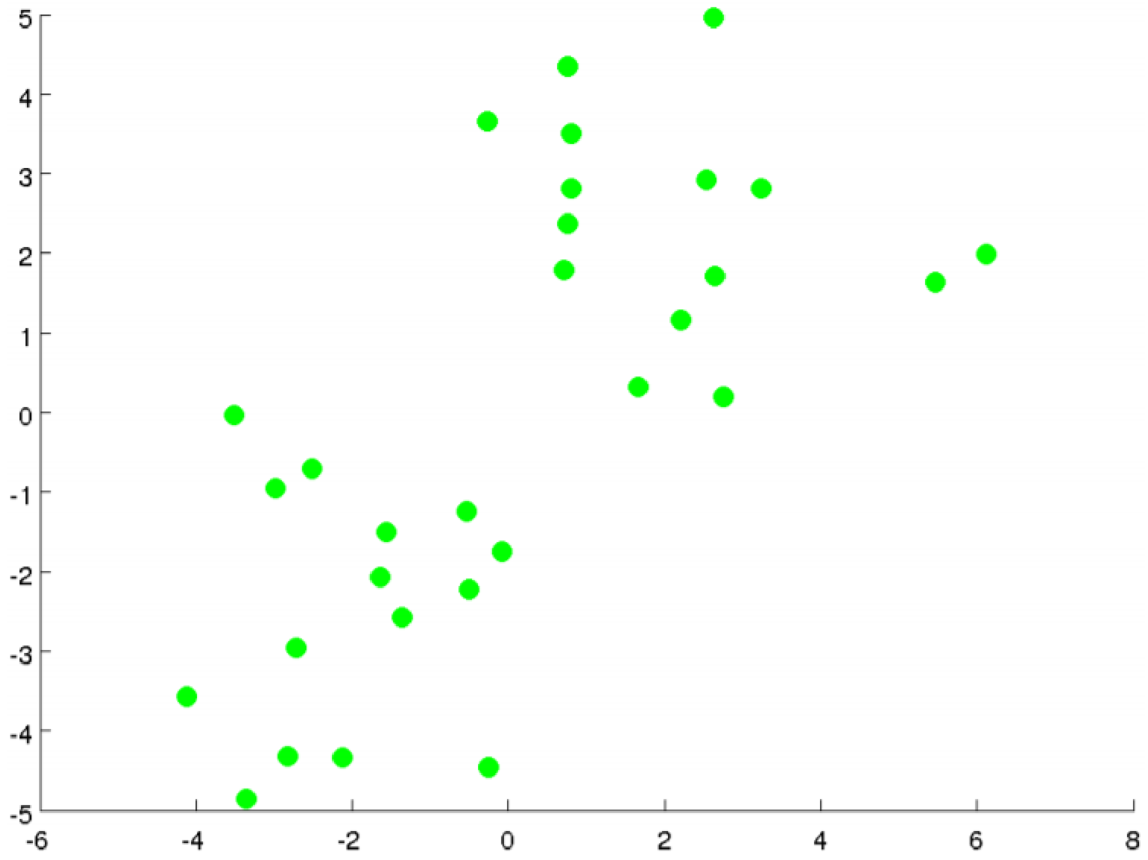
Sigmoid function or Logistic function

Multi-class classification

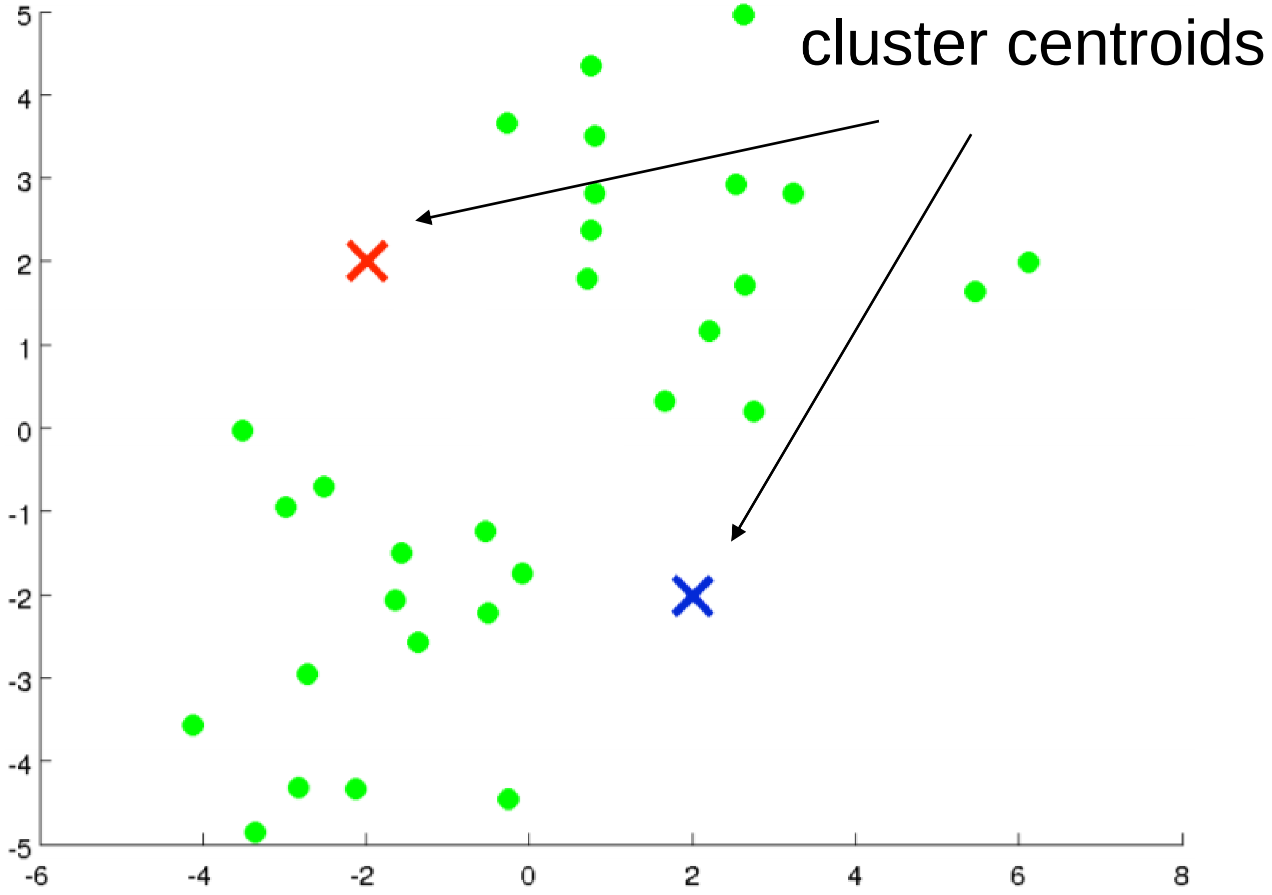
$$h_\theta^i(x) = P(y = i | x; \theta) \ (i \in \{1,2,3\})$$
$$y = prediction = max_i (h_\theta^i(x))$$

Minimizes cost function: mean squared error, ie., incorrect classifications

Clustering – k-means clustering

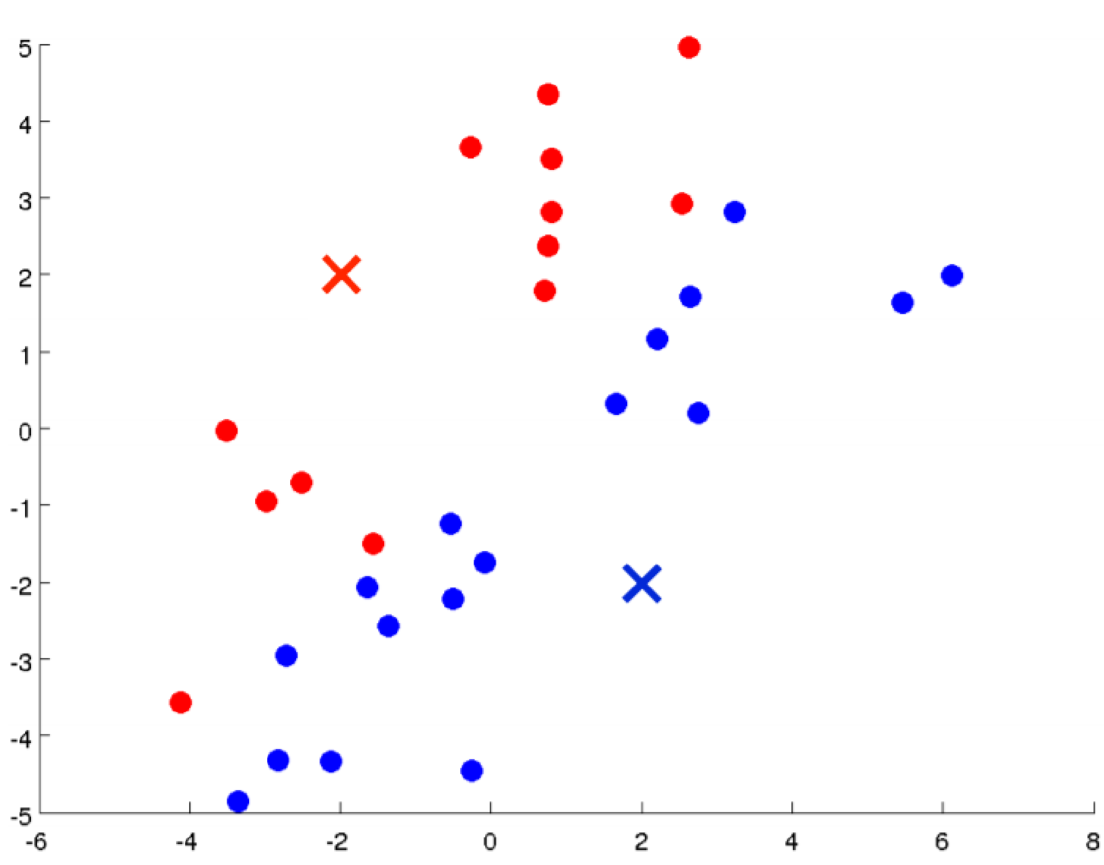


Unclustered or unsegmented data

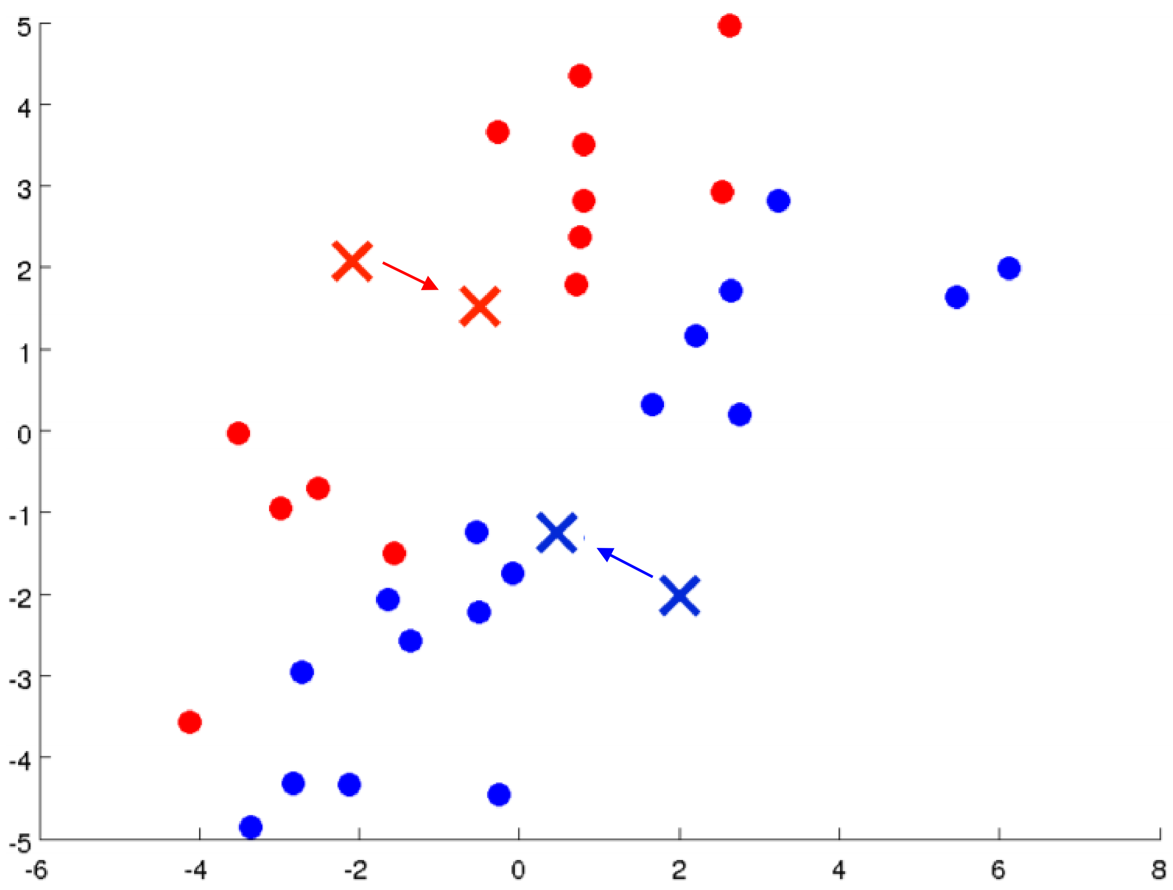


Initial cluster centroids, k=2

k-means clustering contd.

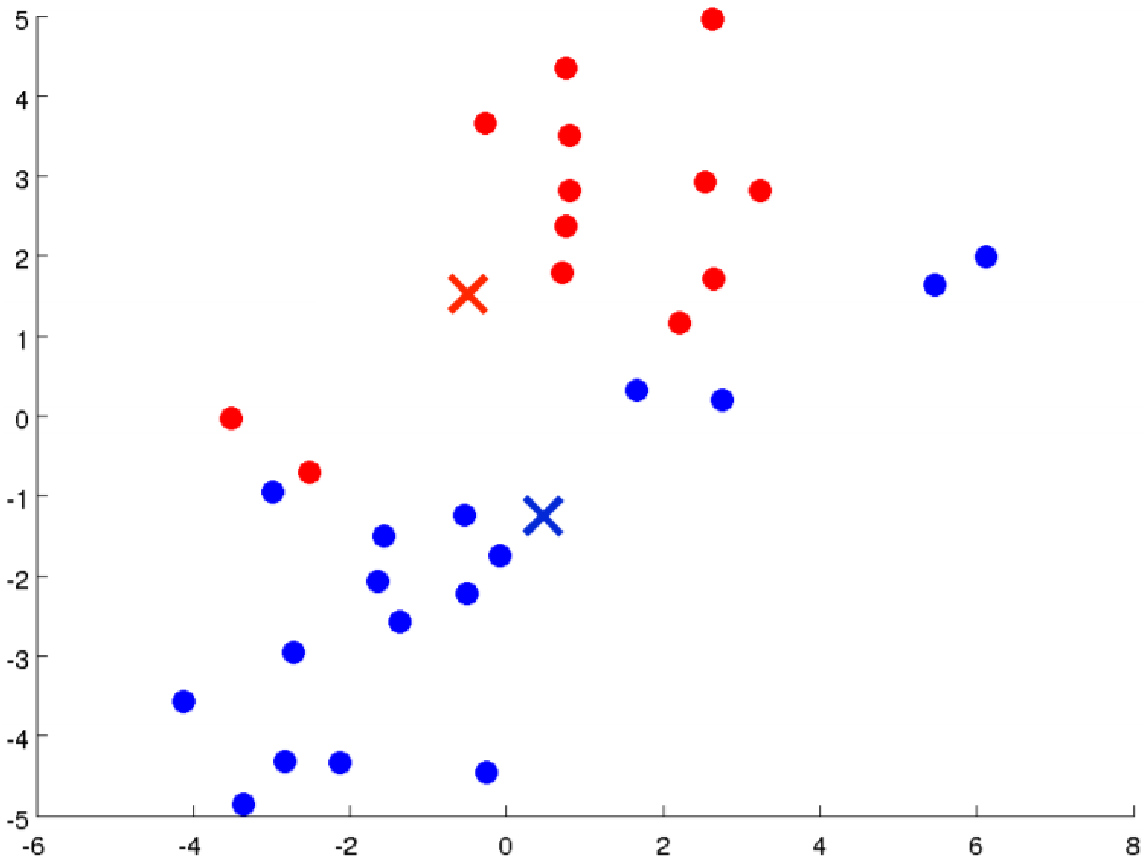


Assign to clusters

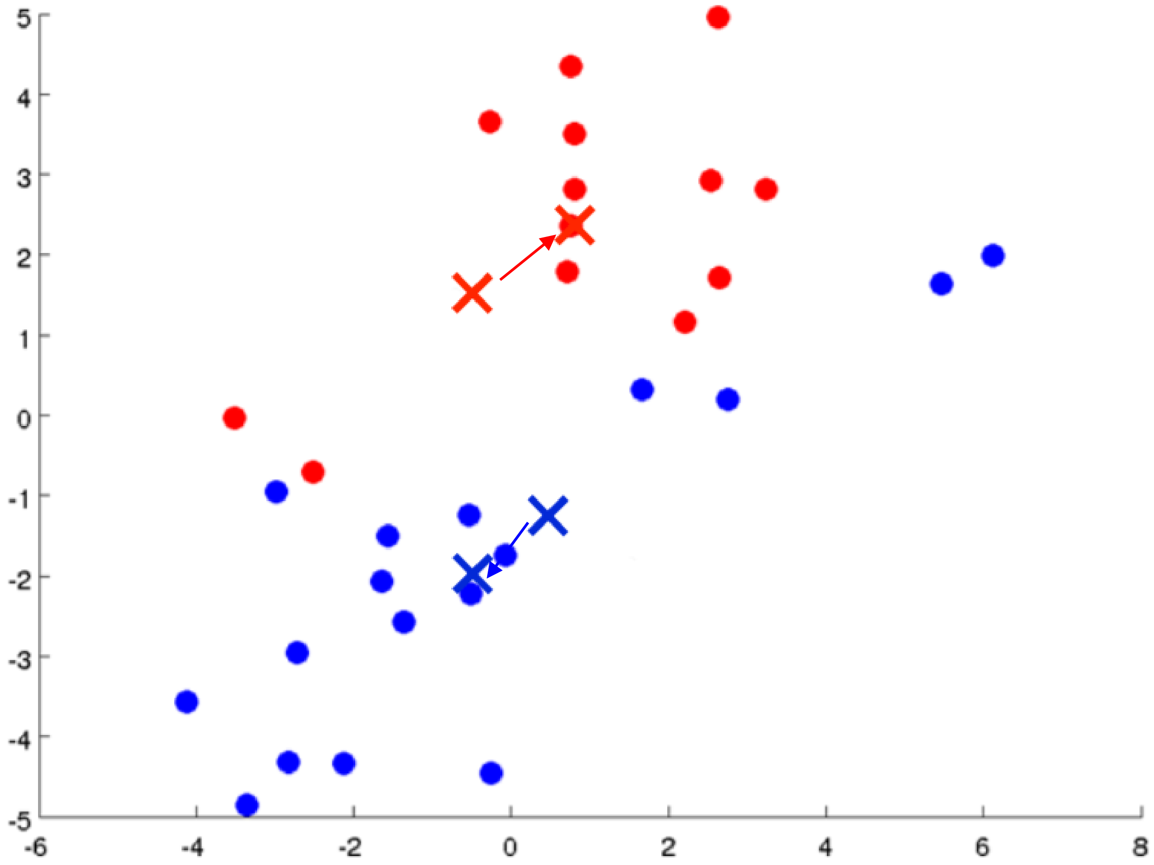


Recalculate / Move the mean

k-means clustering contd.

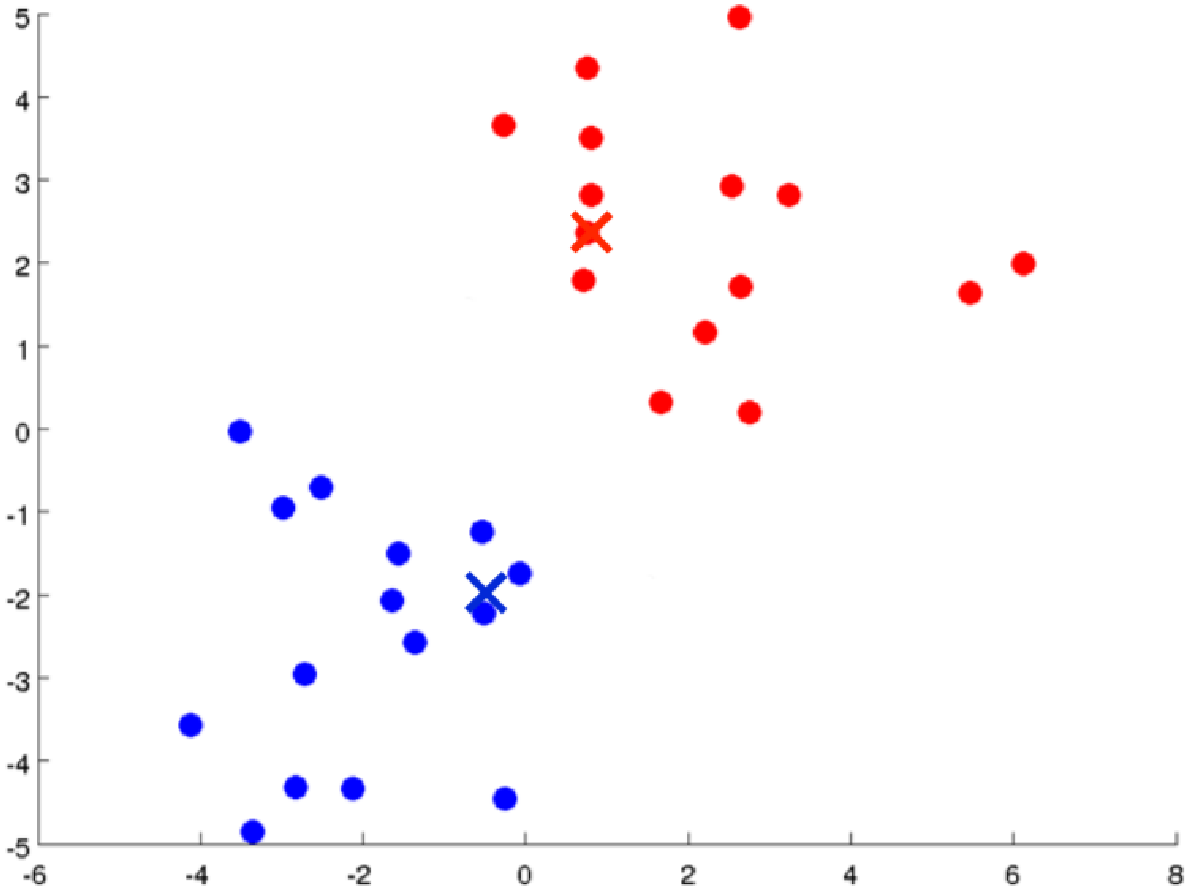


Assign to clusters

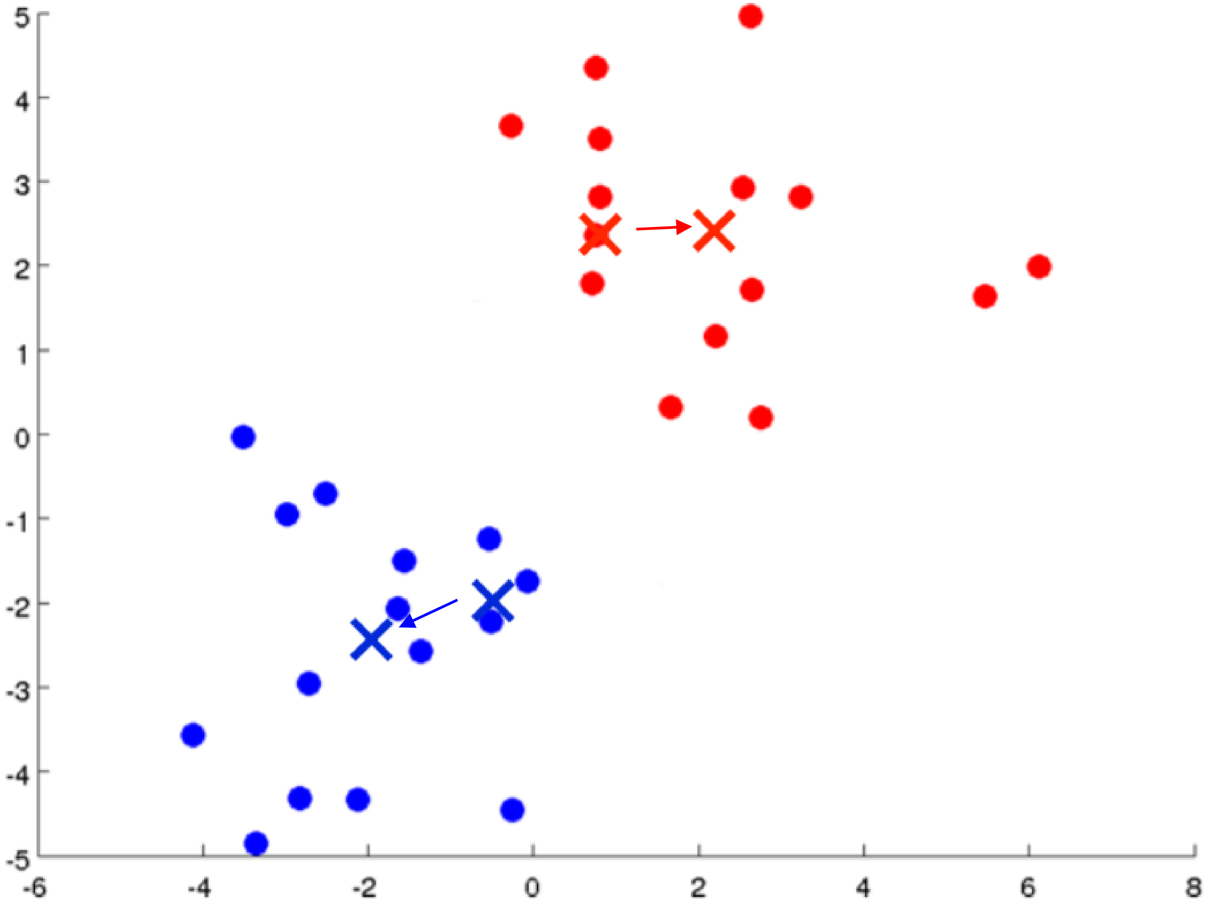


Recalculate / Move the mean

k-means clustering contd.

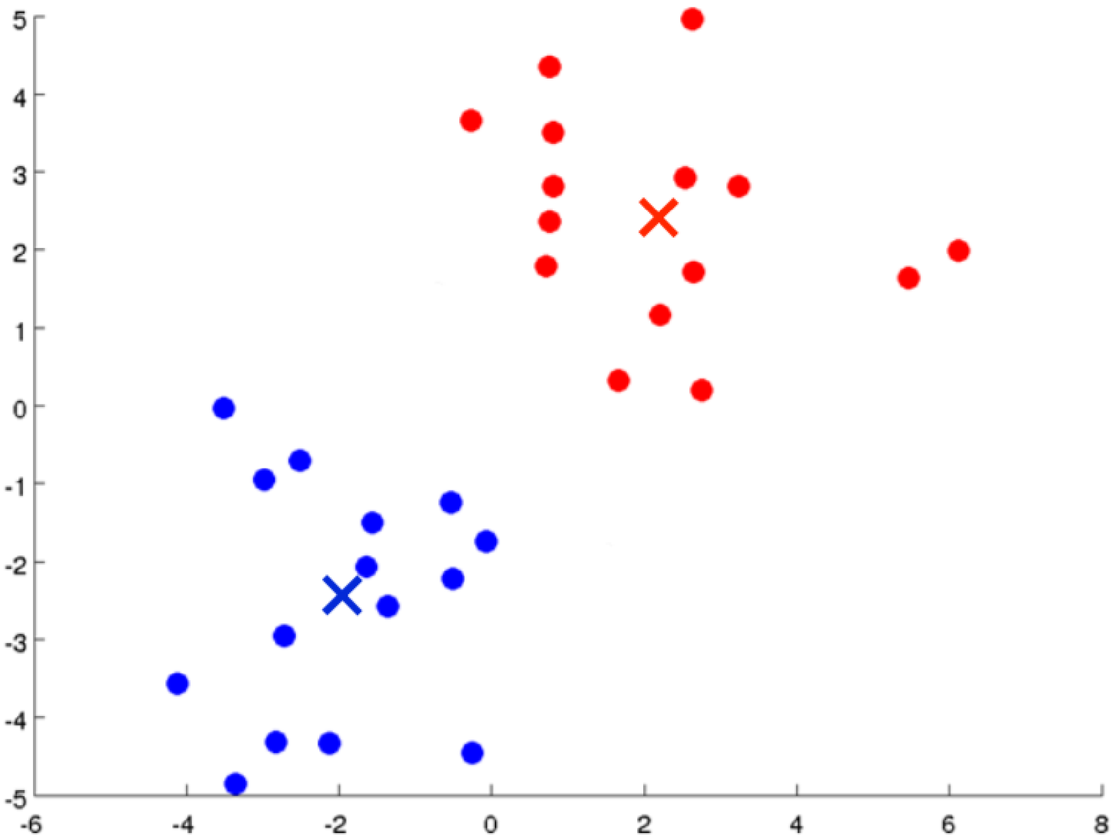


Assign to clusters



Recalculate / Move the mean

k-means clustering contd.



Final clusters & centroids

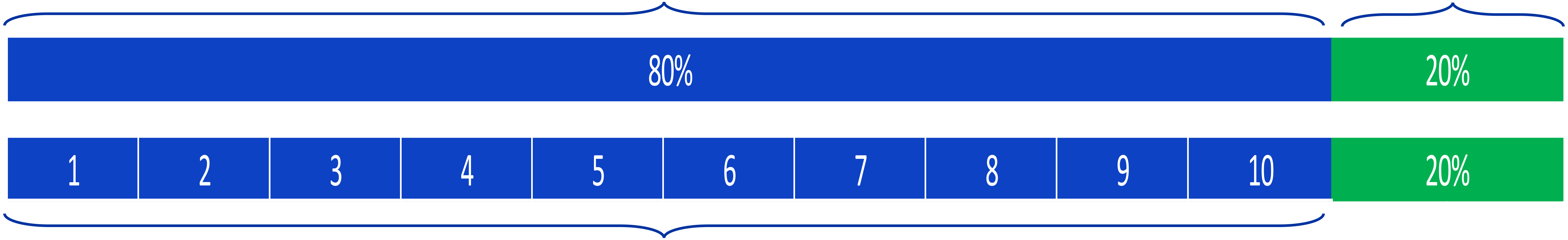
Minimizes cost function:
sum of the distance from cluster means for all examples

What impacts clustering:
- Number of clusters
- Local optima (use random initialization)

Fine Tuning Machine Learning Models

Cross Validation & Testing

- Separate training set and test set at onset and do not use test set for training at any cost
- Use cross validation while testing



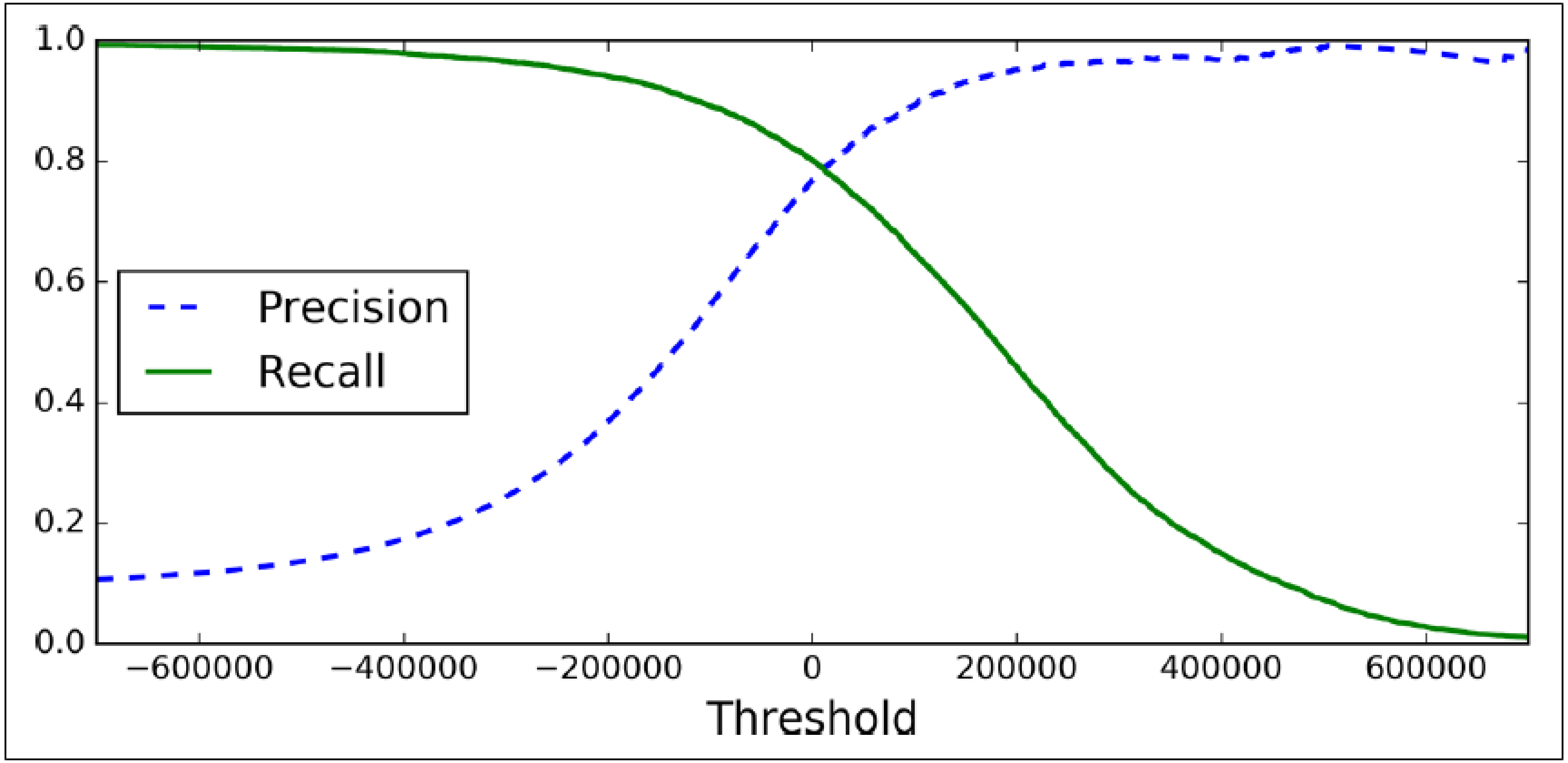
Performance Measures (For Classification)

- Accuracy = Percentage of correct predictions -> Not a good measure
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

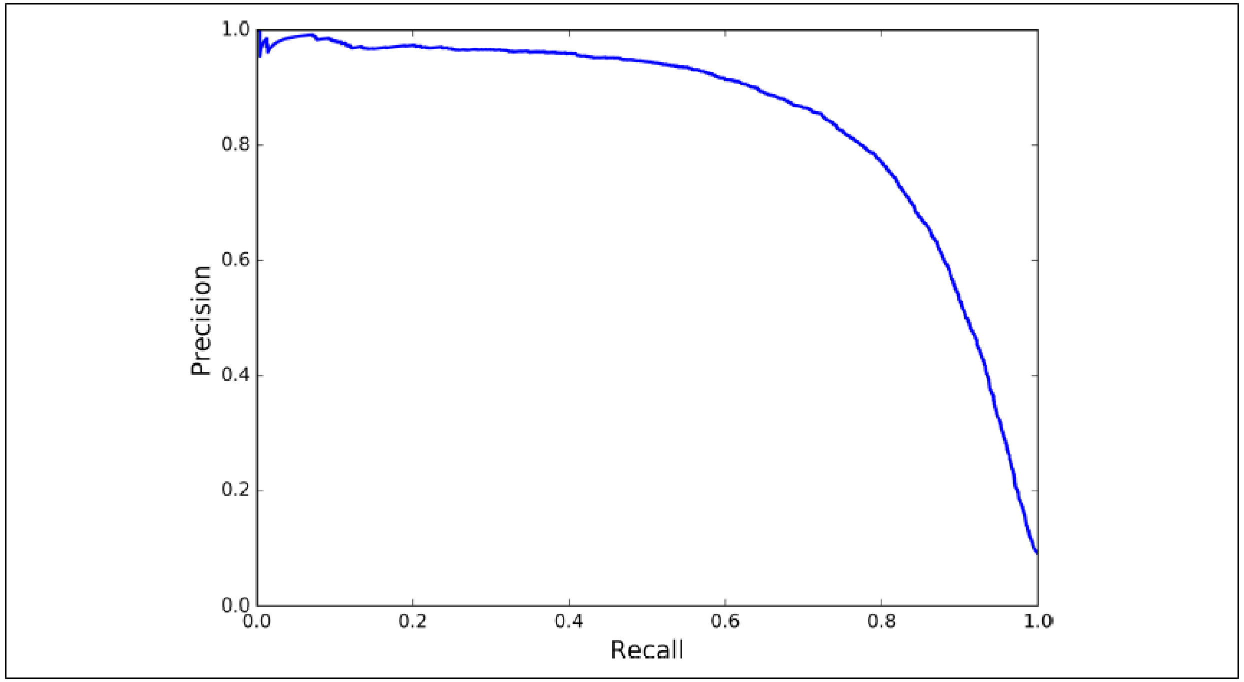
**T = True, F = False, P = Positive, N = Negative*

Confusion Matrix		Actual	
		0	1
Predicted	0	TN	FN
	1	FP	TP

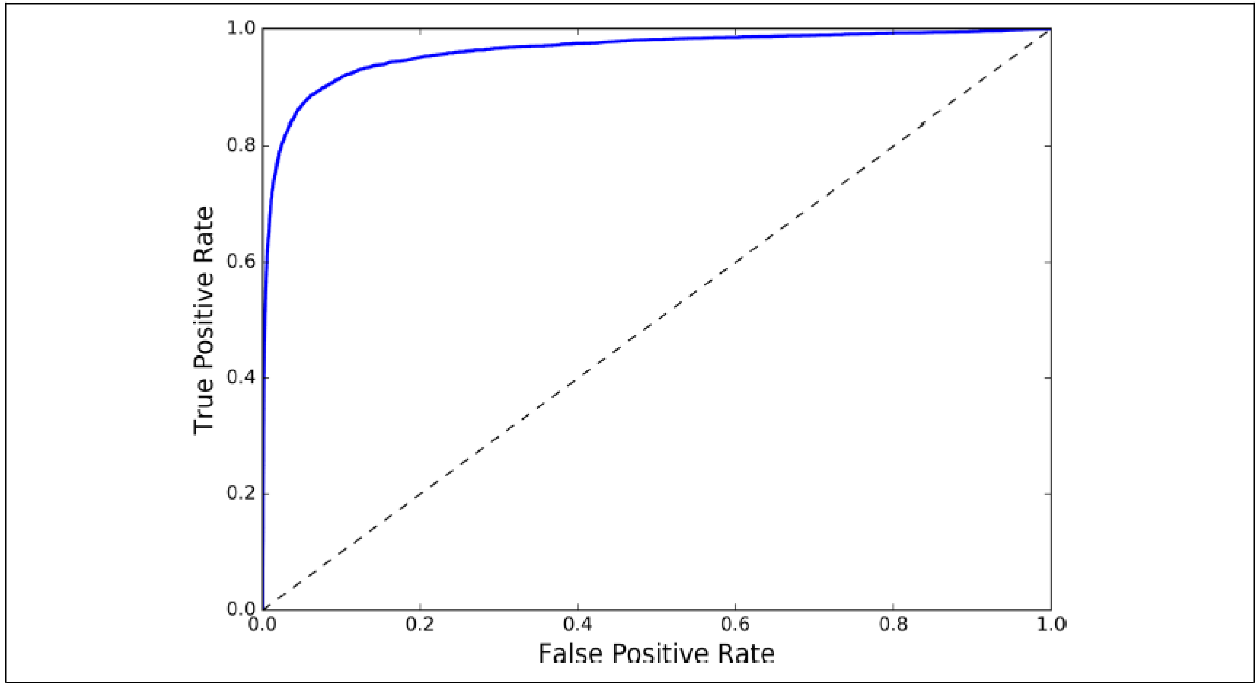
Precision & Recall



Precision & Recall (contd.)



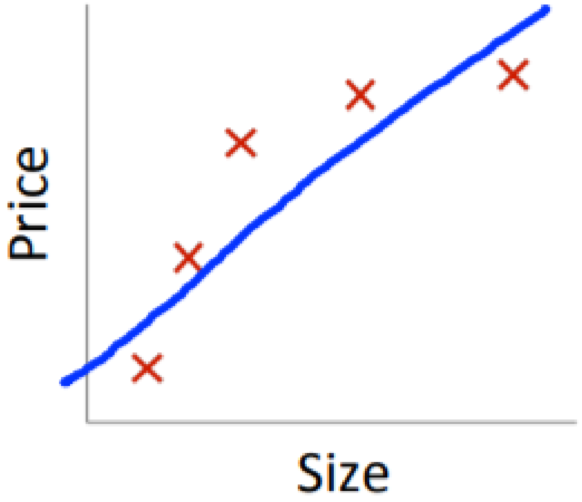
Precision vs Recall



ROC Curve

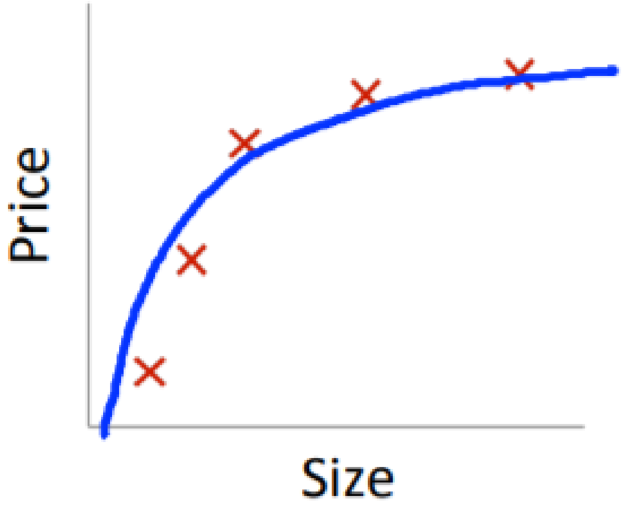
Cognizant

Overfitting & Underfitting



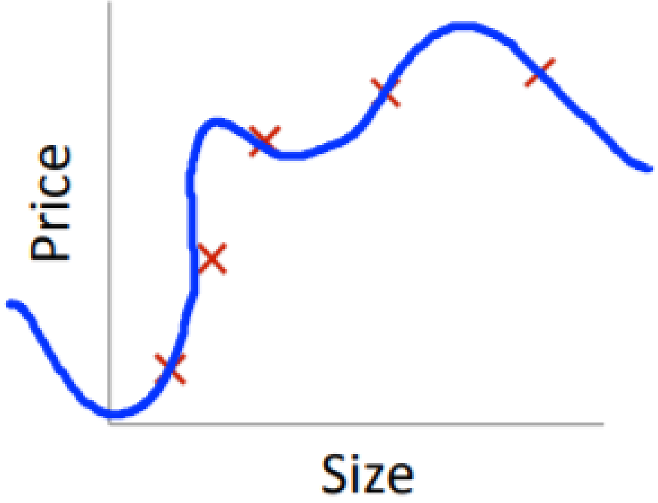
$$\theta_0 + \theta_1 x$$

High Bias
(Underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Just right



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

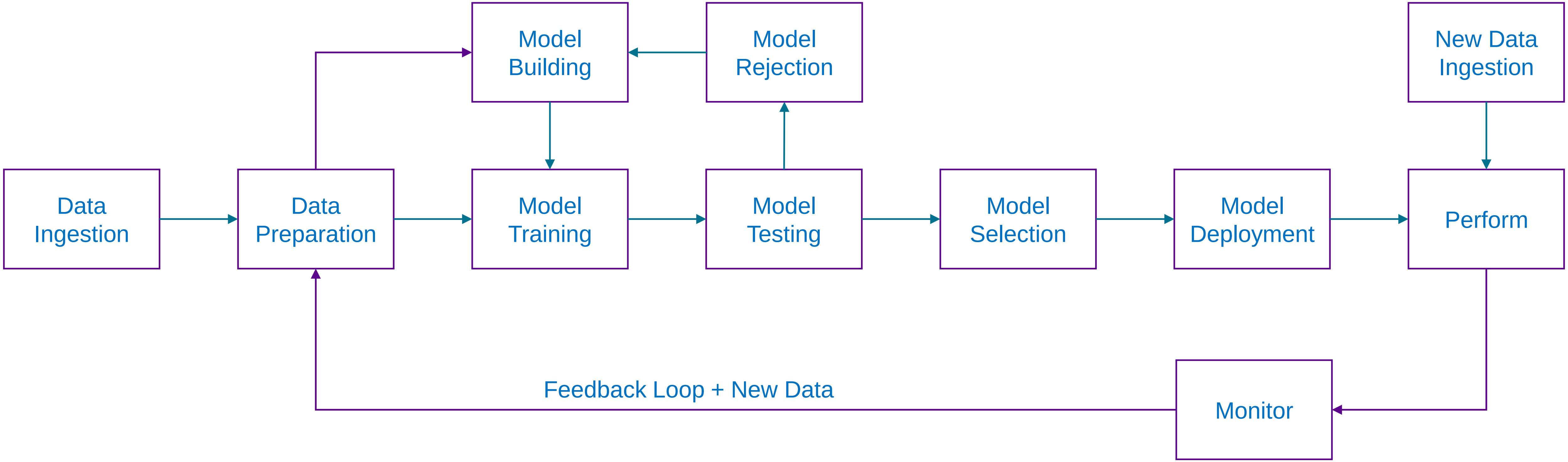
High Variance
(Overfit)

- More powerful model, more parameters
- Better features (feature engineering)
- Reducing the constraints on the model i.e., reduce regularization

- Simpler model, fewer parameters
- Constraining the model through regularization
- Gather more training data
- Reduce the noise in the training data (e.g., fix data errors and remove outliers)

Cognizant

Machine Learning Development Life Cycle



Take Away

- Use the knowledge while building the analytics use cases for LAC
- Understand where to focus – like gather more data or focus on a better model

- AI is the future
- AI and machine learning becoming omnipresent in all spheres of life
- Hope this acts as the spark for more learning!!!

Cognizant

Thank You

Hope it added some value ☺