MACHINE LEARNING

Building your first classifier from scratch

OVERVIEW

- Some definitions
- Types and applications
- Typical workflow/process
- A basic algorithm - KNN (K-Nearest Neighbours) classification
- Performance evaluation
- Have a go yourself

Artificial Intelligence

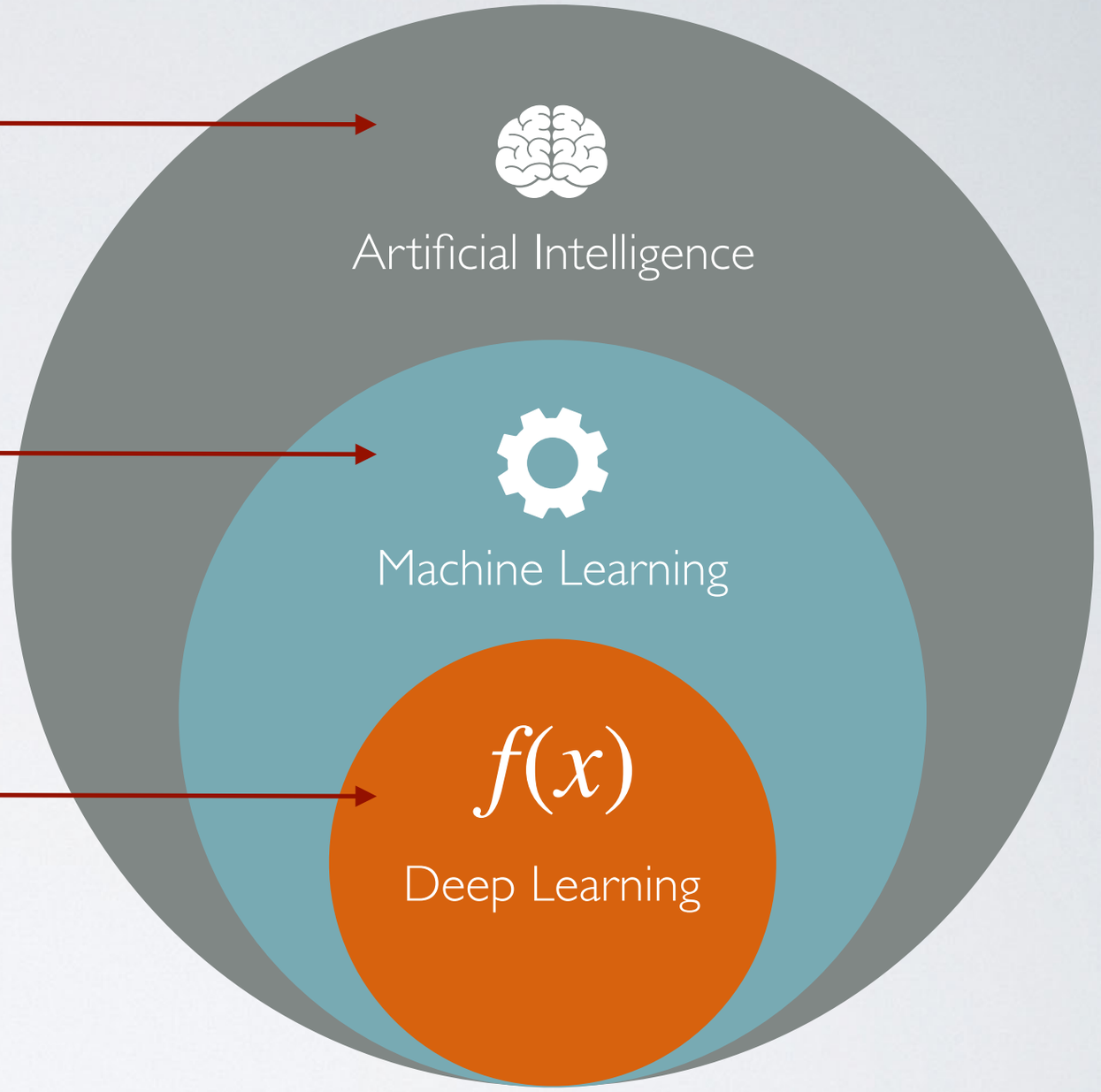
Any technique which enables commuters to mimic human behaviour.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to 'learn' how to carry out tasks without being explicitly programmed how to do them.

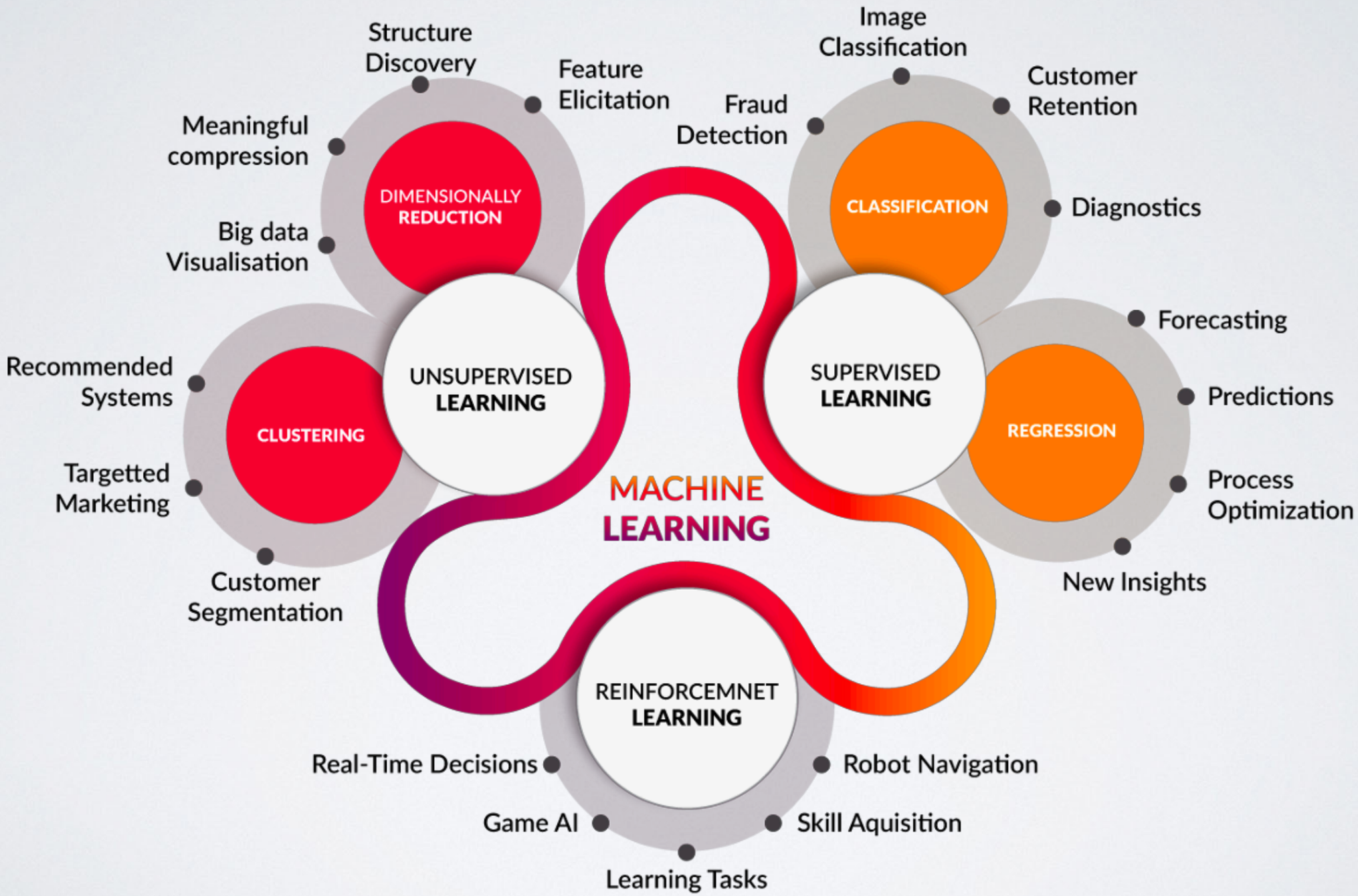
Deep Learning

Subset of ML techniques using multi-layered neural networks based on learning data representations rather than task specific algorithms. They are typically suited to self-learning and feature extraction.

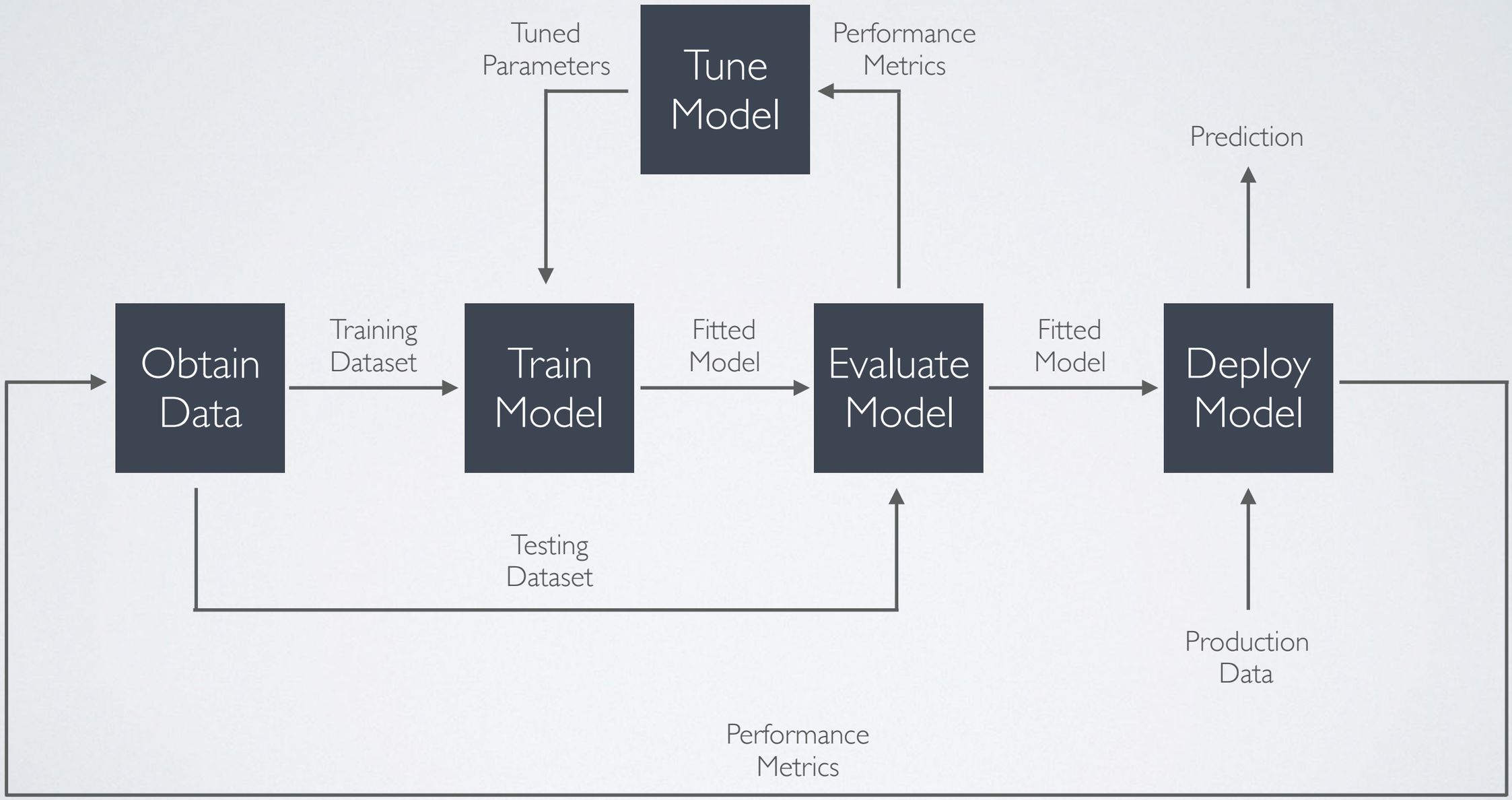


FIELDS OF ARTIFICIAL INTELLIGENCE

MACHINE LEARNING



TYPICAL ML WORKFLOW



THE DIABETES DATASET

- Prima are a group of native Americans living in Arizona
- Highest rate of obesity and diabetes recorded
- Study conducted by National Institute of Diabetes and Digestive and Kidney Diseases collected diagnosis data on female patients with the aim of predicting diabetes.

# Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome (Class Label)
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0

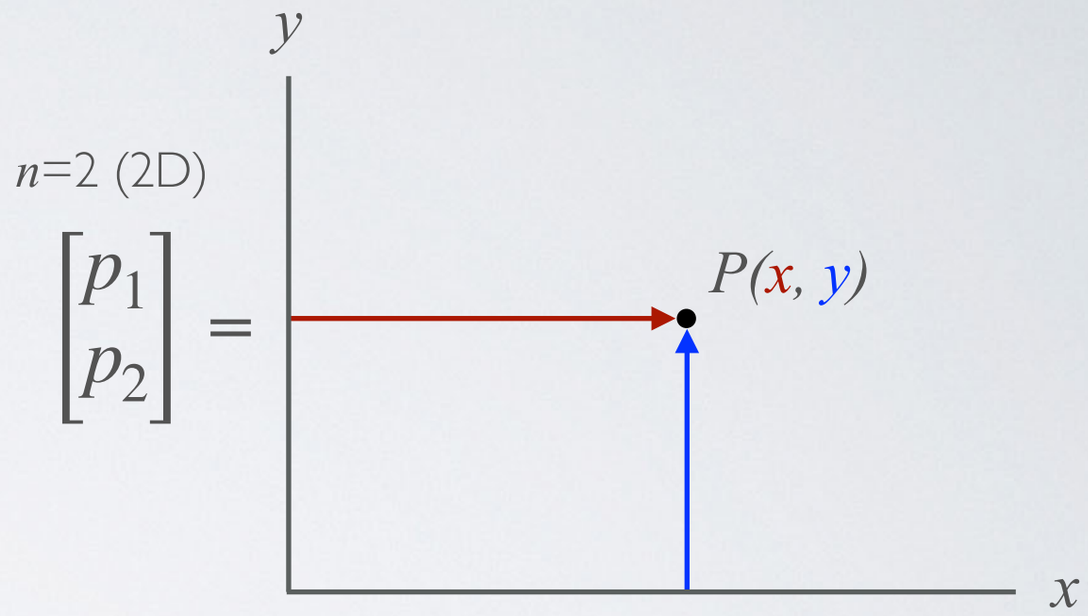
Image Source: Kaggle Prima Indians

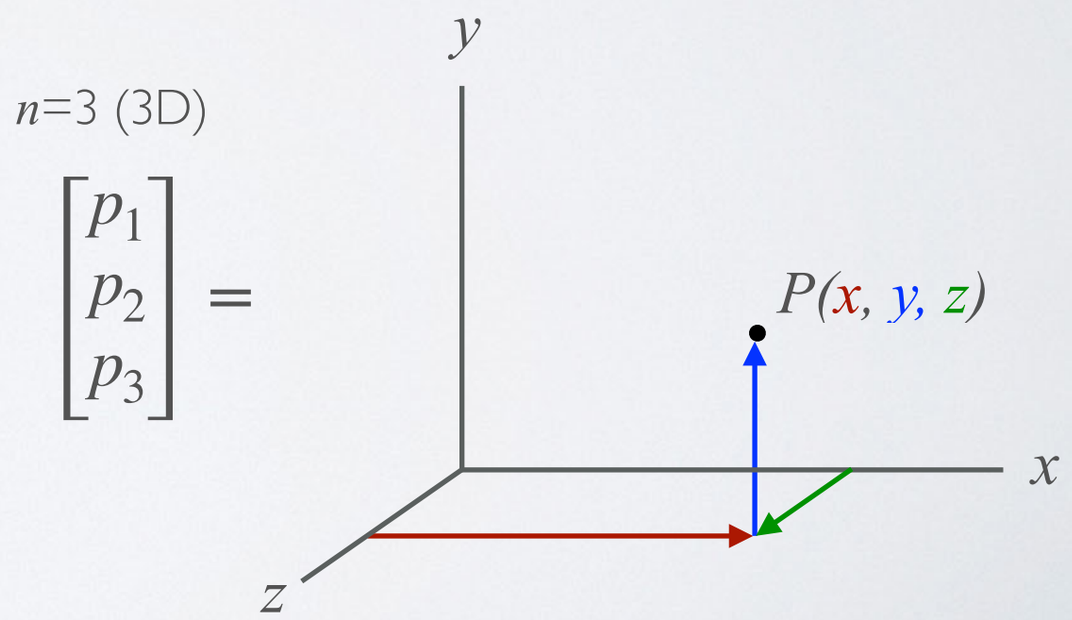
FEATURE VECTORS

- Observations (records) can be represented as n-dimensional numerical feature vectors

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_n \end{bmatrix}$$

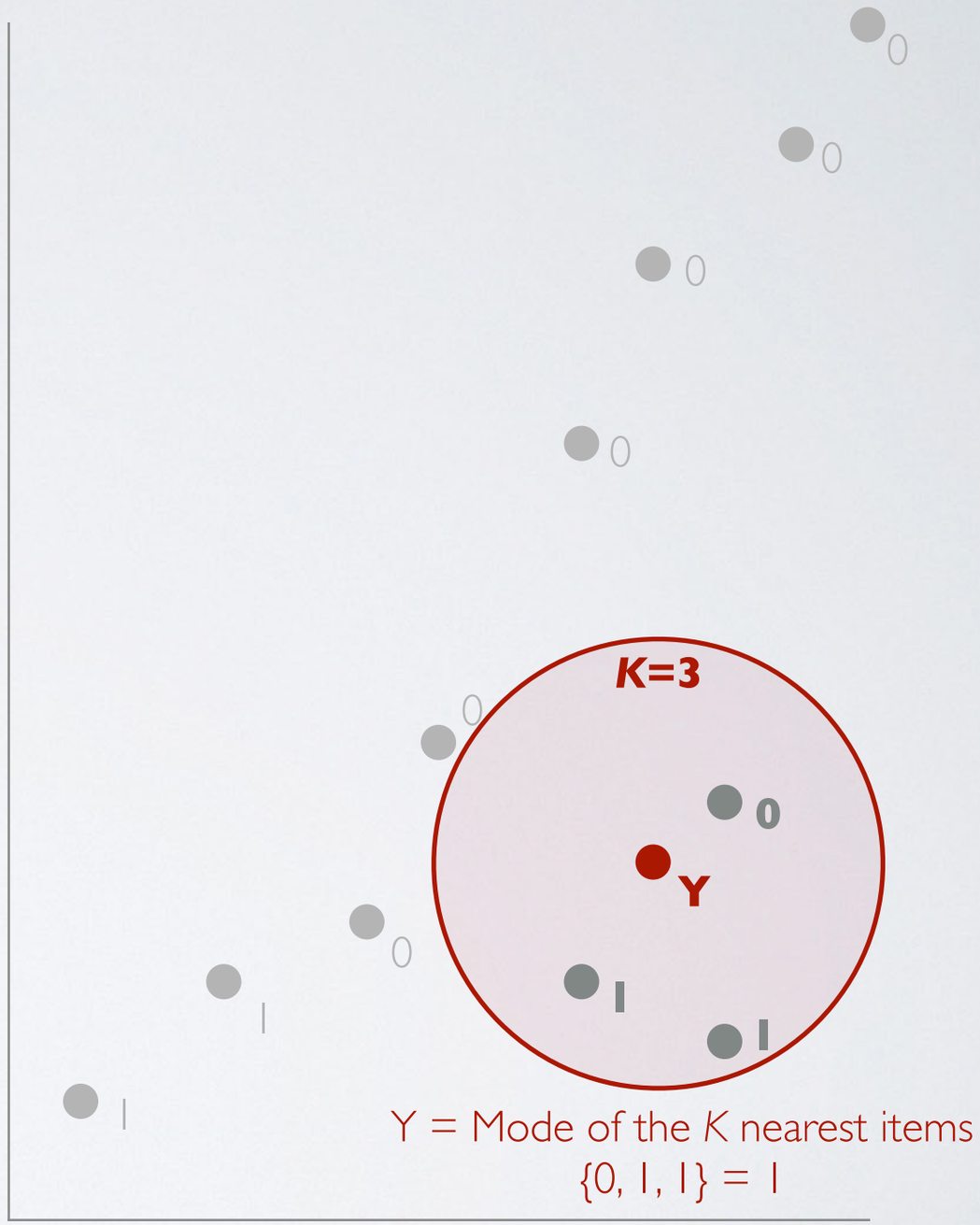
- Feature vectors can be thought of as points in Euclidean space





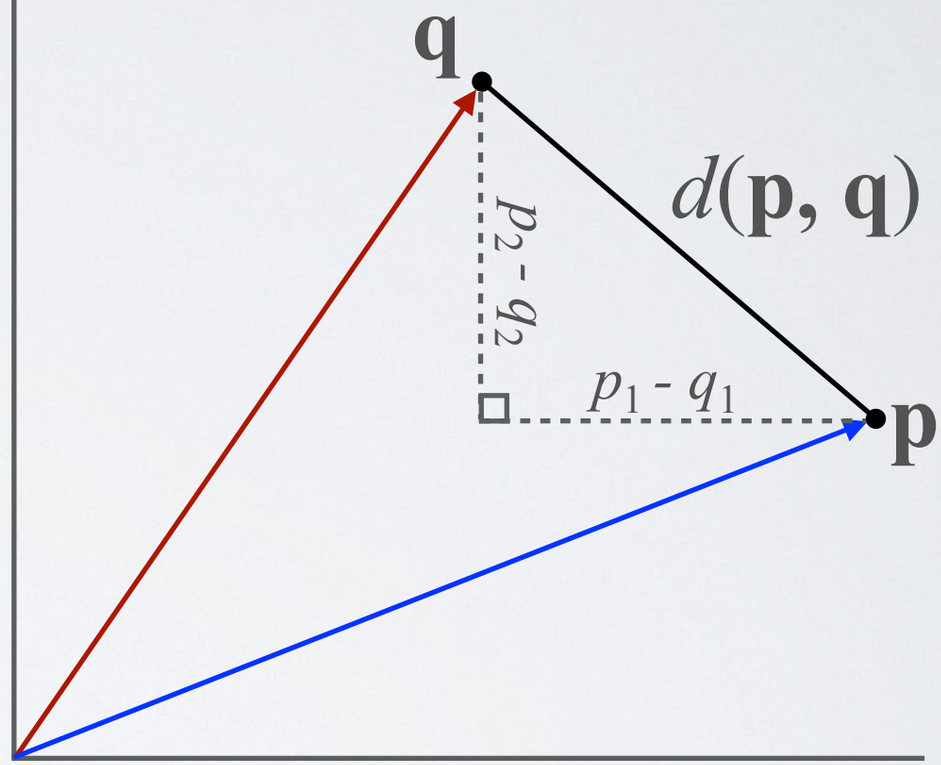
K-NEAREST NEIGHBOURS CLASSIFIER

Predicts class (Y) as the average (mode) of the classes for the *K* nearest neighbours from the training data



NEAREST NEIGHBOURS

- 'Nearest' = shortest distance
- Where distance uses a formal distance metric
- In n dimensional Euclidean space, distance between points p and q is given by Pythagoras formula:



$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

$$= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

EVALUATING PERFORMANCE

The Confusion Matrix

Total Observations (n)		Actual		
		Yes	No	
Predicted	Yes	True Positives	False Positives	$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
	No	False Negatives	True Negatives	
$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$		$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$		$\text{accuracy} = \frac{\text{TP} + \text{TN}}{n}$

GETTING STARTED

1. Get the evaluation harness and dataset in JS or Go:

   http://github.com/fresh8/mlworkshop

2. Build your model:

```
type Predictor interface {
   Fit(X *mat.Dense, Y []string)
   Predict(X *mat.Dense) []string
}
```

3. Evaluate your model with the harness:

```
result, err := harness.Evaluate("diabetes.csv", &model)
```

4. Share your result (F1 score) via Slack and we will update the leaderboard:

   https://bit.ly/2TO0FEz

QUESTIONS