

# Machine Learning



## Machine Learning

Lecture: Instance-Based Learning

Ted Scully

# Instance Based Learning

- ▶ Instance-based learning is a family of learning algorithms that **compare new problem instances with existing instances in the training data.**
- ▶ Predictions for new instances are based on their **similarity to stored instances** (the basis of the similarity measure is typically distance)

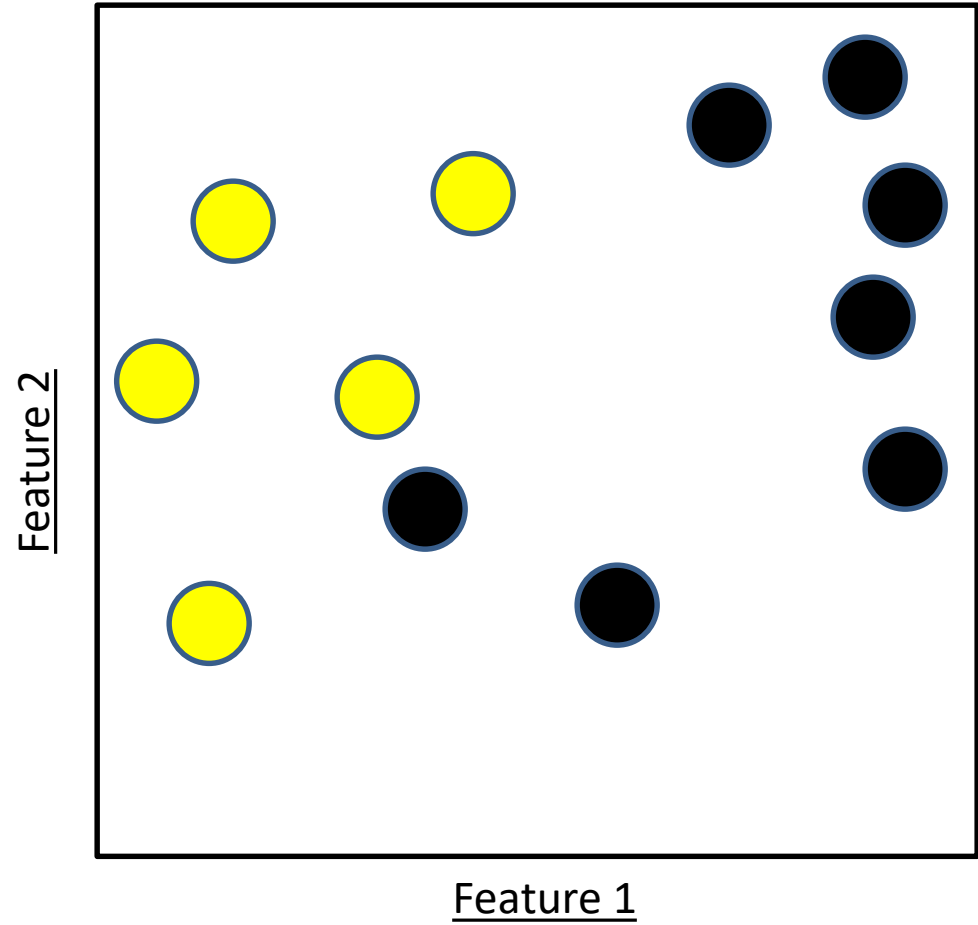
# Nearest Neighbour Algorithm (1)

- ▶ The Nearest Neighbour algorithm is the simplest form of IBL
- ▶ Nearest Neighbour algorithm:
  - ▶ Given a test case with a value to be predicted, identify which stored case it is nearest.
  - ▶ Assigns the **new test case the same class as the nearest neighbour**
  - ▶ Requires a distance metric.
- ▶ This very simple algorithm is very susceptible to noise.

Given a query instance  $\mathbf{x}_q$ ,  
first locate the nearest training example  $\mathbf{x}_n$   
then  $\mathbf{f}(\mathbf{x}_q) := \mathbf{f}(\mathbf{x}_n)$

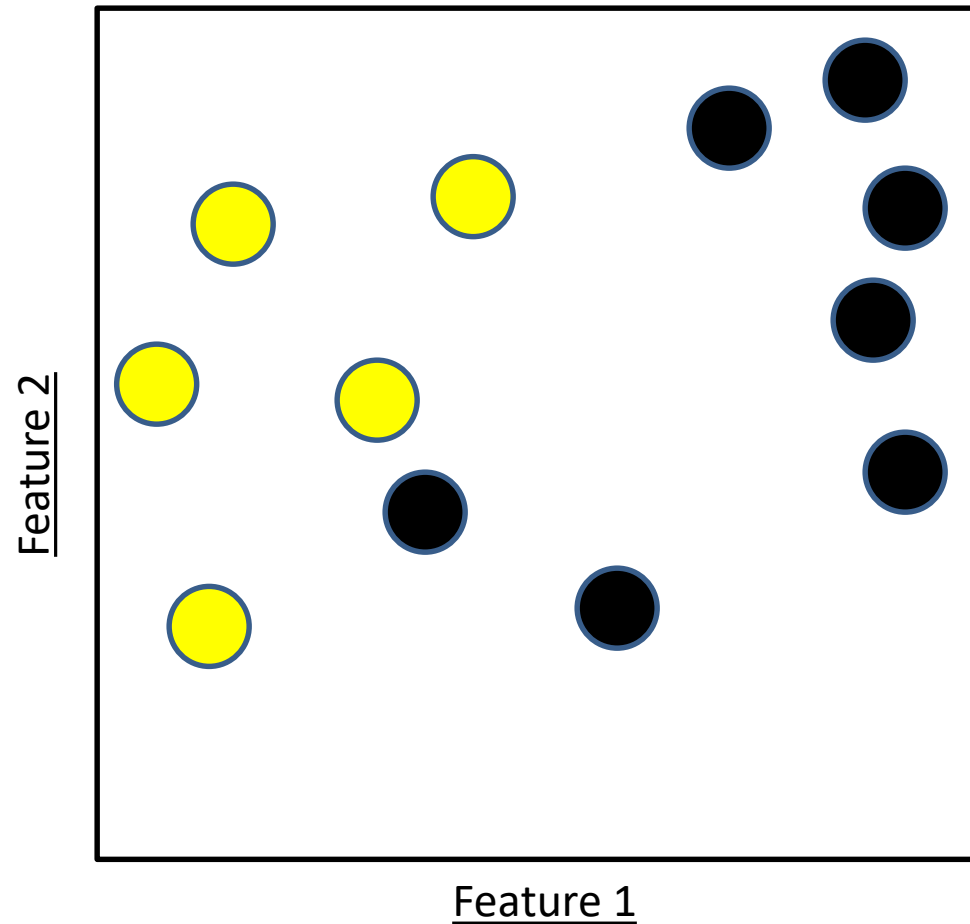
Where  $\mathbf{f}(\mathbf{x}_n)$  is the class associated with the data item  $\mathbf{x}_n$

Feature 1	Feature 2	Colour



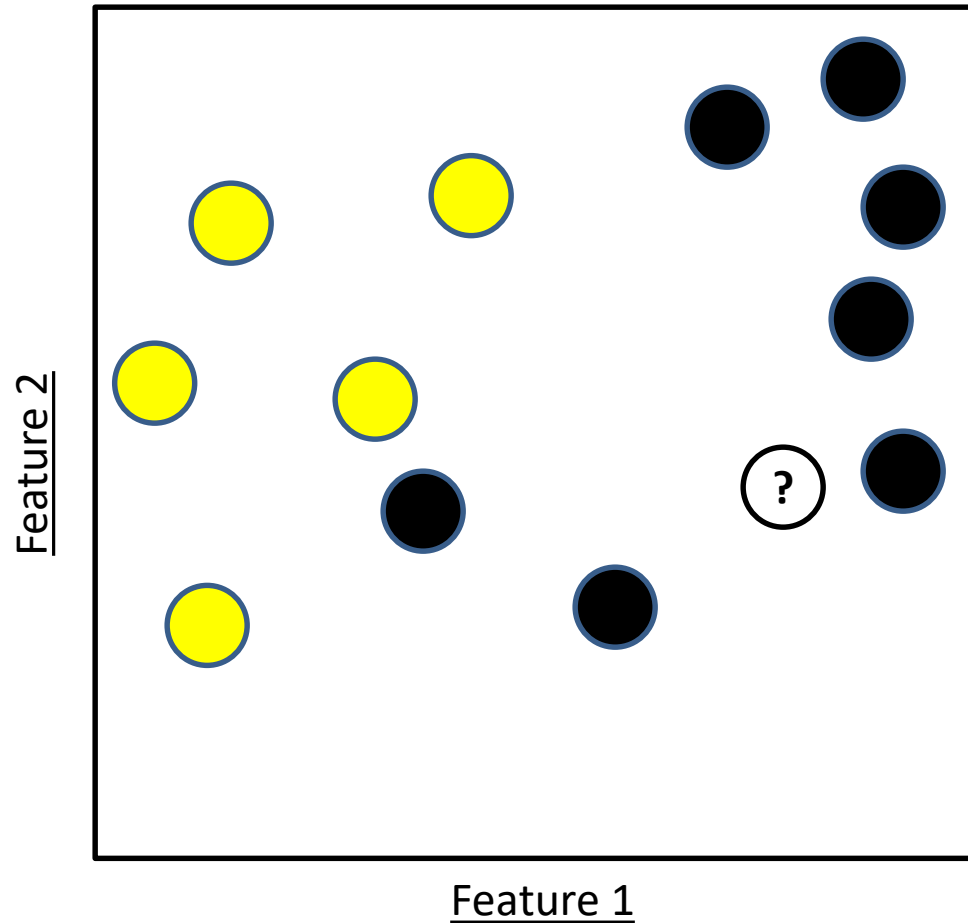
Feature 1	Feature 2	Colour

- ▶ We can represent a dataset in an IBL by mapping all instances to a **feature space**, that is, using each descriptive feature as an axis of a coordinate system.
- ▶ We can then place each instance within the feature space based on the value of it's features.

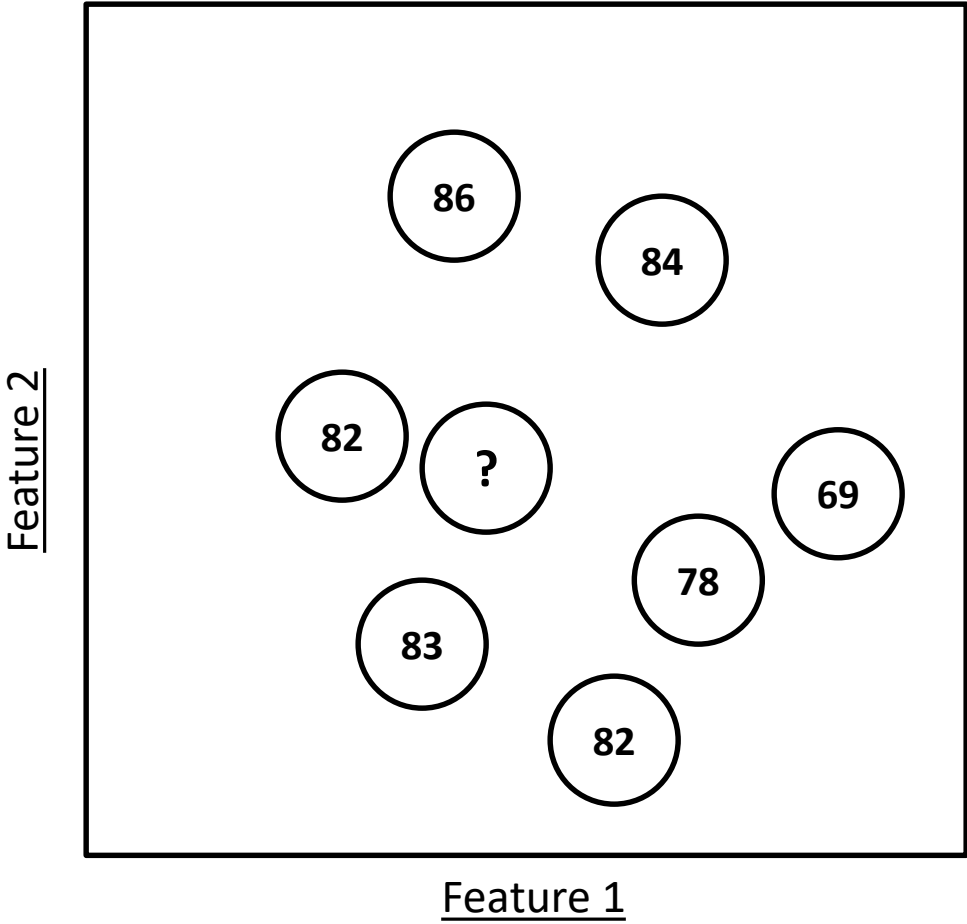


# Nearest Neighbour Example

- ▶ We wish to classify the new case, which is the white circle with the question mark.
- ▶ The nearest neighbour in feature space is selected and the example instance is assigned the same class.



Feature 1	Feature 2	Regression Target



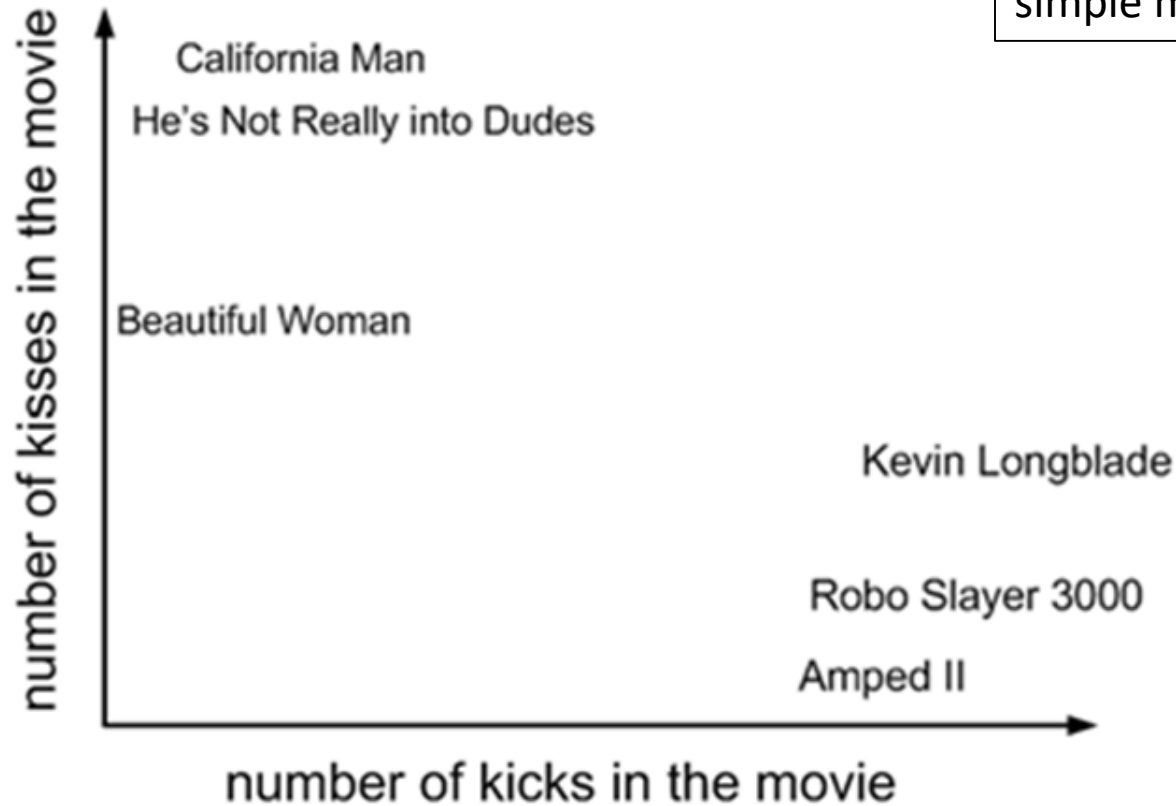
# Sample Dataset

Movie title	# of kicks	# of kisses	Type of movie
<i>California Man</i>	3	104	Romance
<i>He's Not Really into Dudes</i>	2	100	Romance
<i>Beautiful Woman</i>	1	81	Romance
<i>Kevin Longblade</i>	101	10	Action
<i>Robo Slayer 3000</i>	99	5	Action
<i>Amped II</i>	98	2	Action



# Feature Space

Here we can see the feature space for our simple movie dataset.

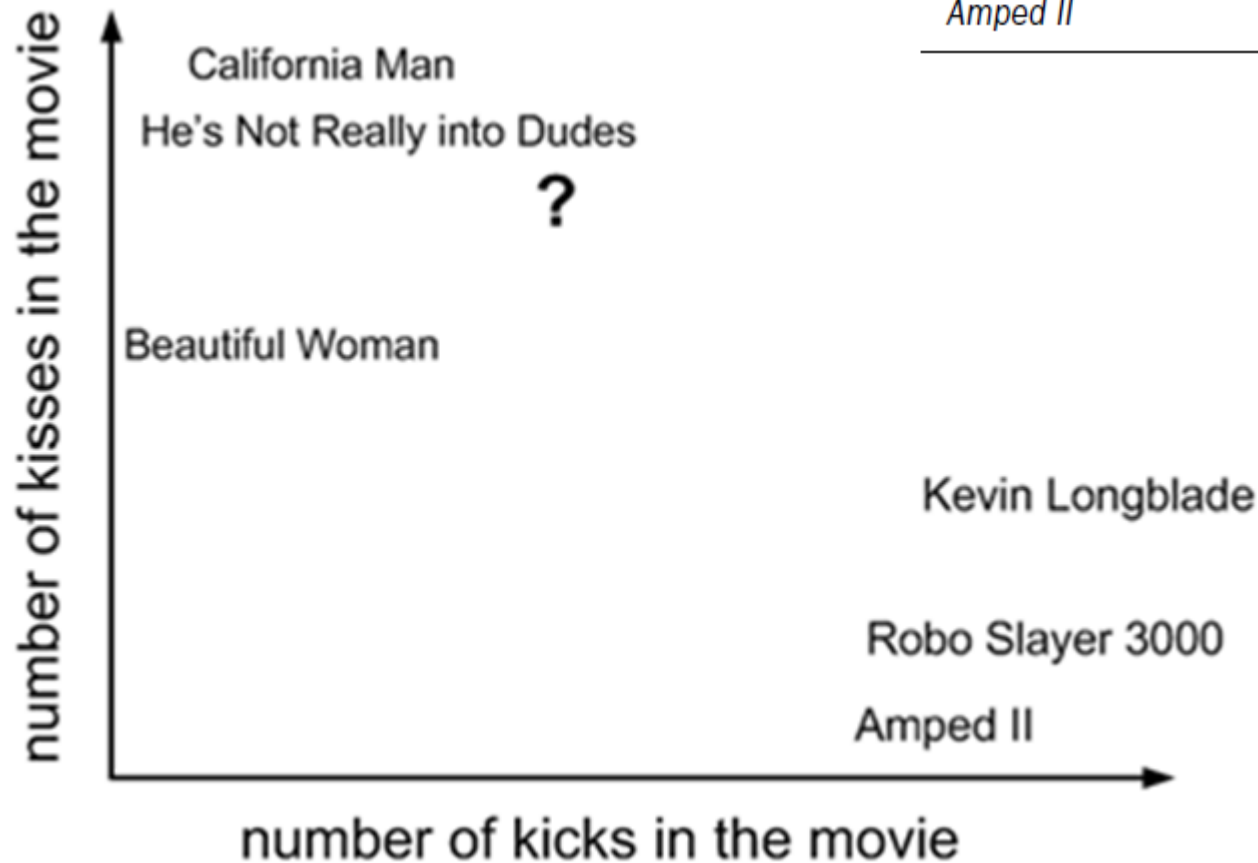


Movie title	# of klcks	# of klsses	Type of movle
<i>California Man</i>	3	104	Romance
<i>He's Not Really into Dudes</i>	2	100	Romance
<i>Beautiful Woman</i>	1	81	Romance
<i>Kevin Longblade</i>	101	10	Action
<i>Robo Slayer 3000</i>	99	5	Action
<i>Amped II</i>	98	2	Action
?	18	90	Unknown

Assume we get an unseen movie and we have to classify it as a Romance or action based on it's feature values.

Given a query instance  $\mathbf{x}_q$ ,  
 first locate the nearest  
 training example  $\mathbf{x}_n$   
 then  $\mathbf{f}(\mathbf{x}_q) := \mathbf{f}(\mathbf{x}_n)$


Movie title	Distance to movie “?”
California Man	20.5
He’s Not Really into Dudes	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9



As the query is  
 closest to the  
 film “He’s Not  
 Really into  
 Dudes” then it is  
 classified as a  
**Romance**

Movie title	# of kicks	# of kisses	Type of movie
<i>California Man</i>	3	104	Romance
<i>He's Not Really into Dudes</i>	2	100	Action
<i>Beautiful Woman</i>	1	81	Romance
<i>Kevin Longblade</i>	101	10	Action
<i>Robo Slayer 3000</i>	99	5	Action
<i>Amped II</i>	98	2	Action
?	18	90	Unknown

Noise: An incorrect classification



What would happen if “He’s no really into dudes” was incorrectly classified as an Action movie.

Our new query instance ‘?’ would also get **incorrectly classified** as an Action.

The simple nearest neighbour approach is very likely to over-fit on the training data.

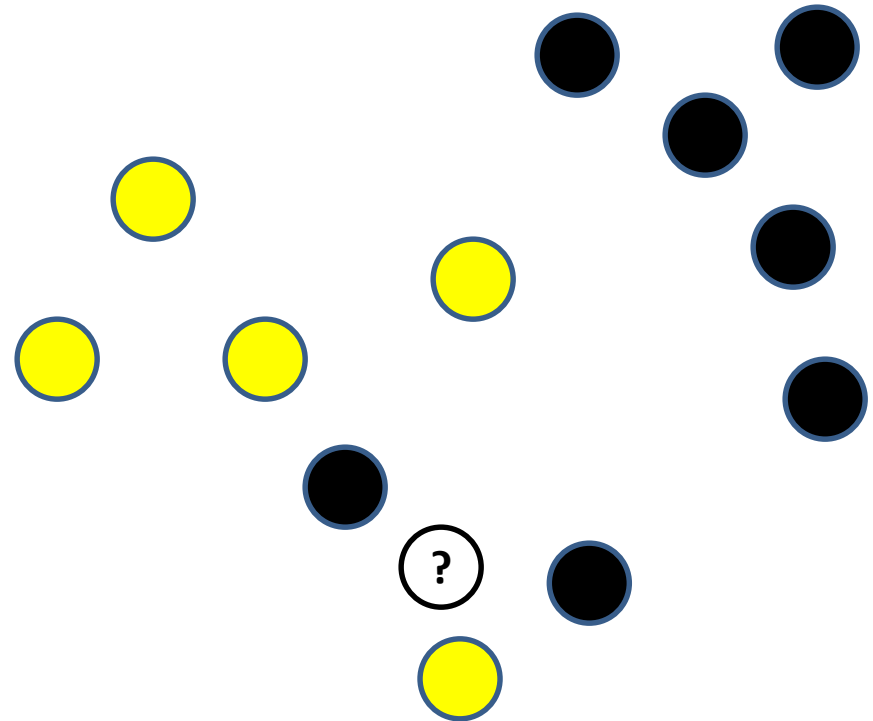
Any ideas of how we might go about improving the algorithm?

# K-Nearest Neighbour

- ▶ A simple extension is to consider not just the nearest neighbour, but **several nearest neighbours**.
- ▶ This requires defining a neighbourhood; the standard approach is to use a neighbourhood that is just large enough to include a fixed number of points,  $k$ .
- ▶ Prediction is based on these  $k$  nearest neighbours.
  - ▶ If this is a **regression** problem then use the **average** of  $k$ -nearest neighbours.
  - ▶ If it is a **classification** problem then take a **vote** amongst the  $k$ -nearest neighbours
  - ▶ This approach is less sensitive to noise.

# k - Nearest Neighbour Algorithm - Classification

- ▶ k-NN can be applied to classification problems
- ▶ If it is a classification problem then take a **majority vote** amongst the k-nearest neighbours
- ▶ What is the classification of the query instance if  $k = 3$ ? What if  $k = 5$



# k - Nearest Neighbour Algorithm - Regression

We assume a set of training examples  $\langle x_i, f(x_i) \rangle$

Given a query instance  $x_q$ ,

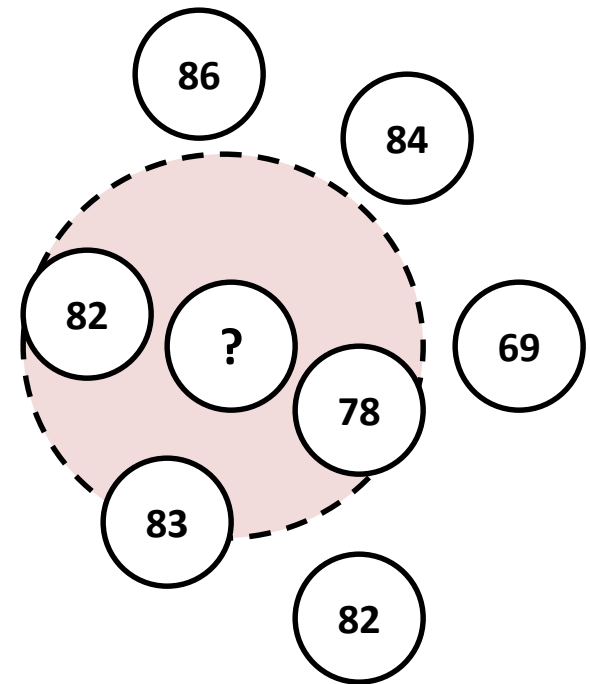
Identify  $k$  nearest training examples

If it is regression problem, then average values of  $k$ -nearest neighbours

$$f(x_q) := \frac{\sum_{i=1}^k f(x_i)}{k}$$

# k - Nearest Neighbour Algorithm - Regression

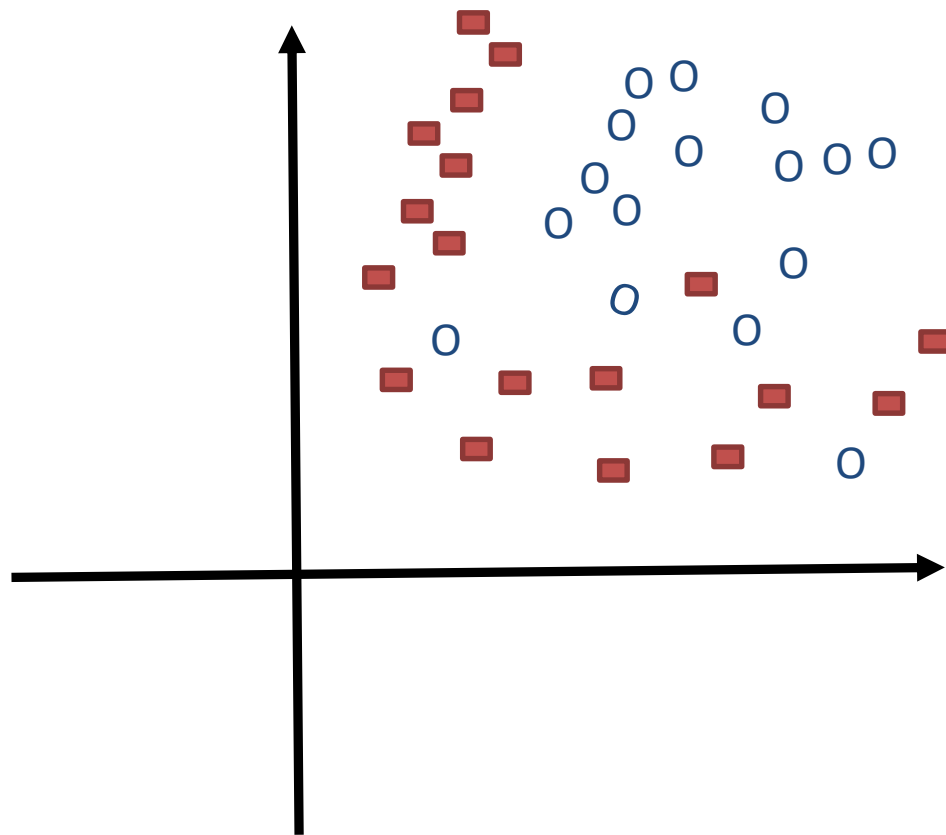
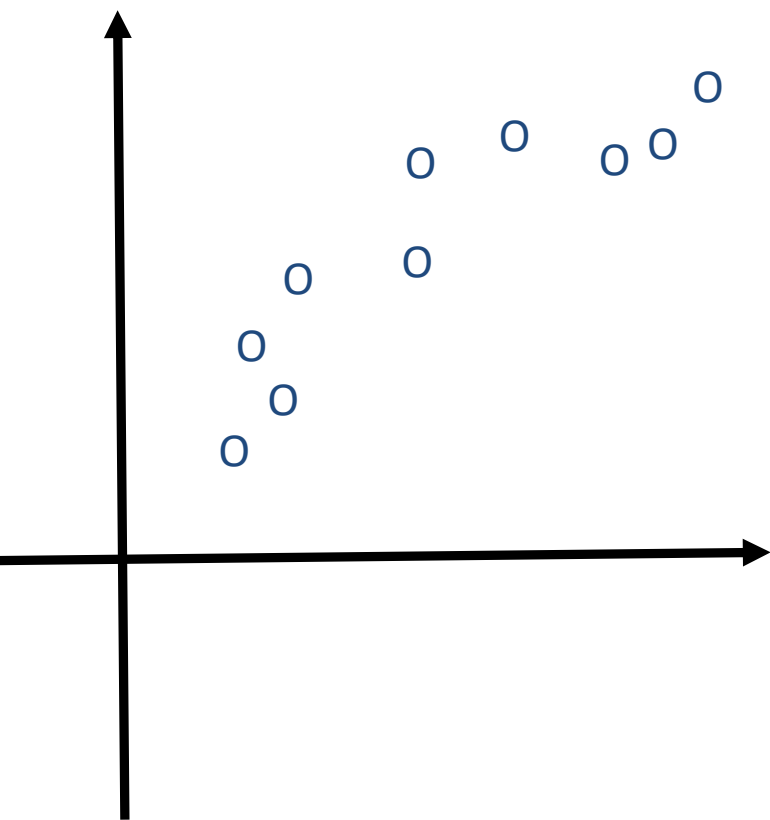
- ▶ k – NN can be applied to regression problems
- ▶ The answer is the average value of k of the neighbours
- ▶ What is the answer if:  $k = 3$ ?  
 $(82+78+83)/3 = 81.$

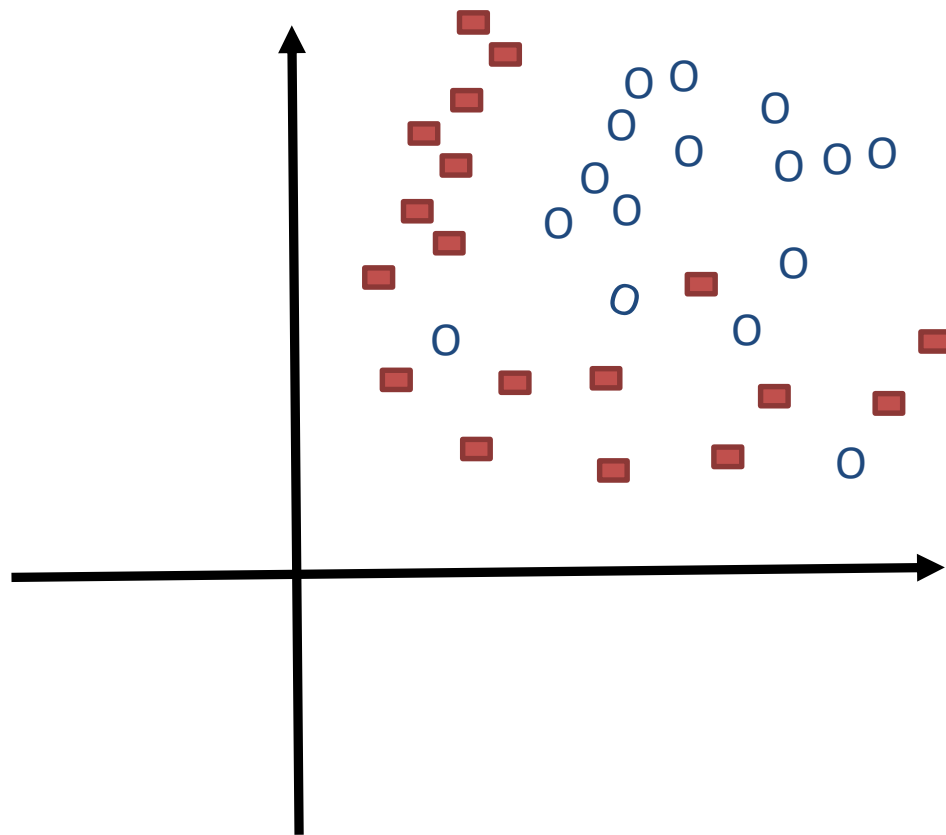
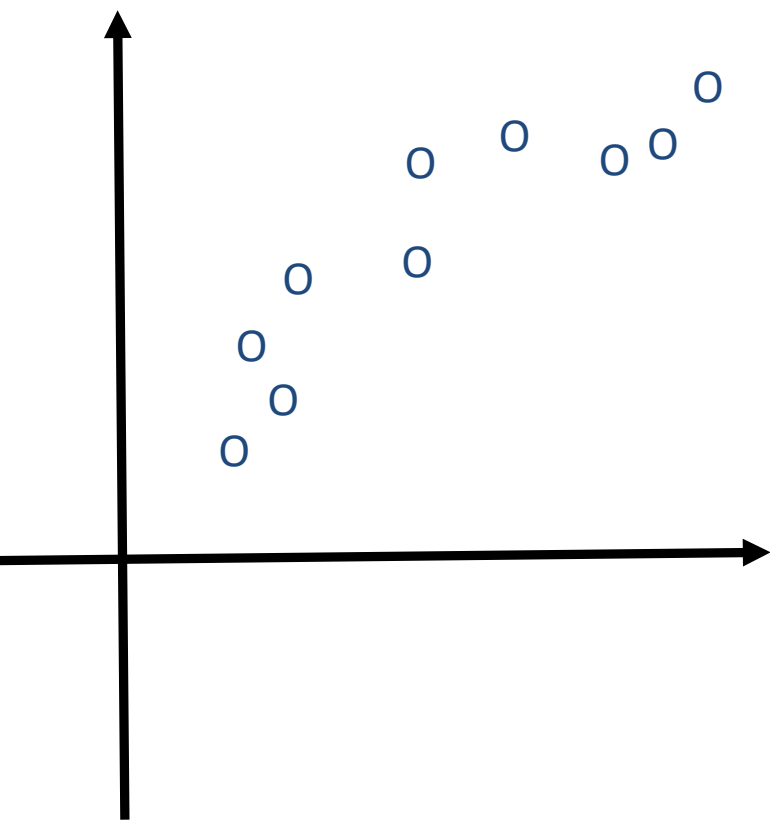


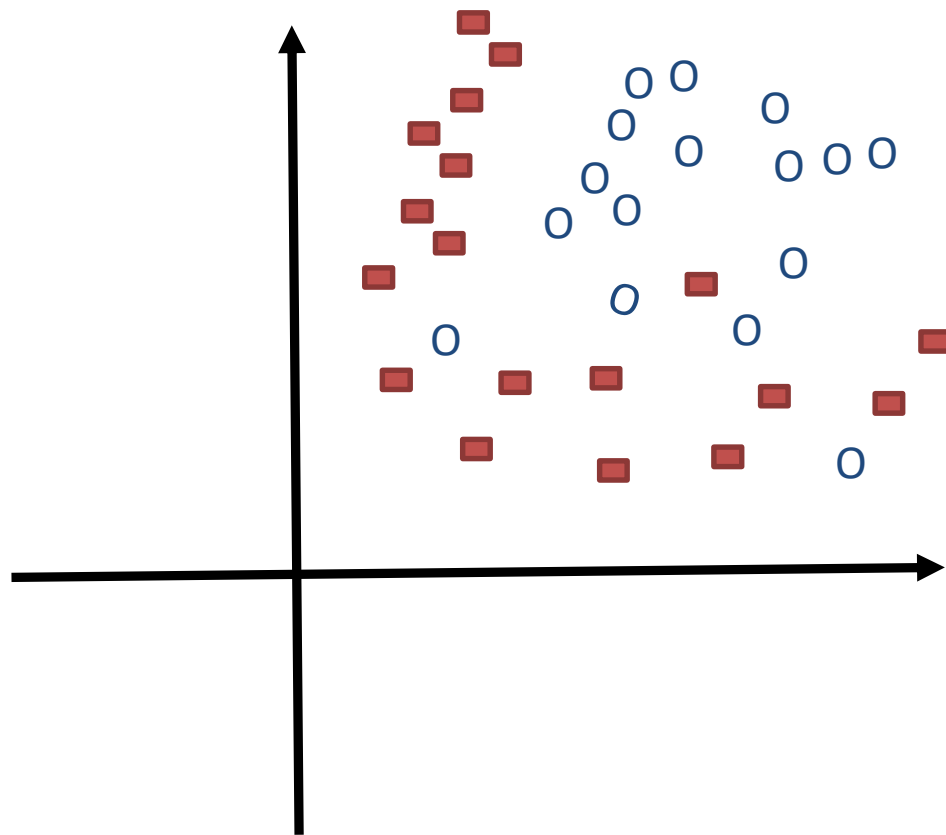
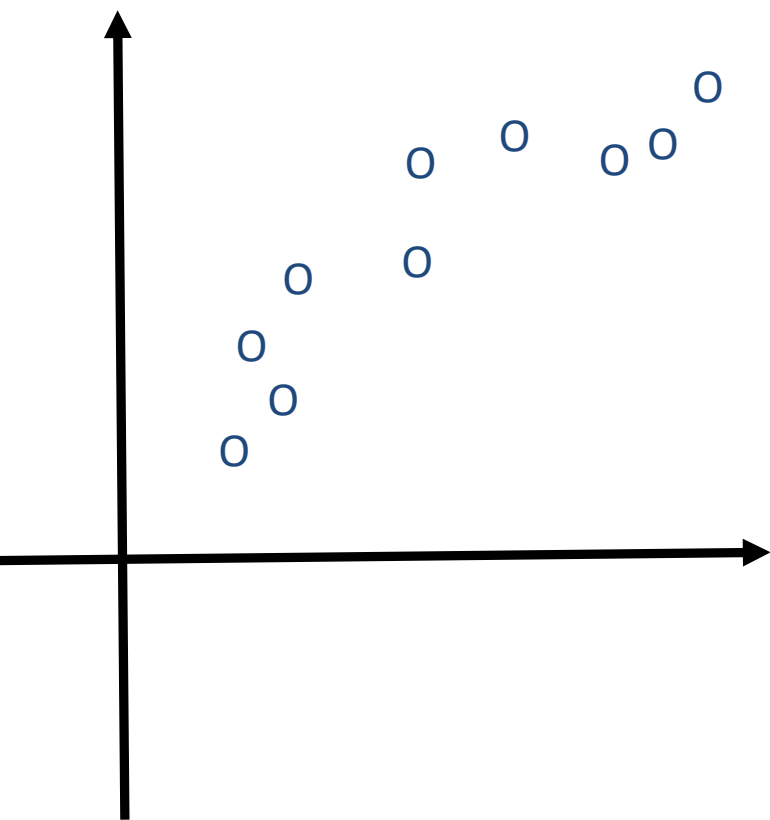


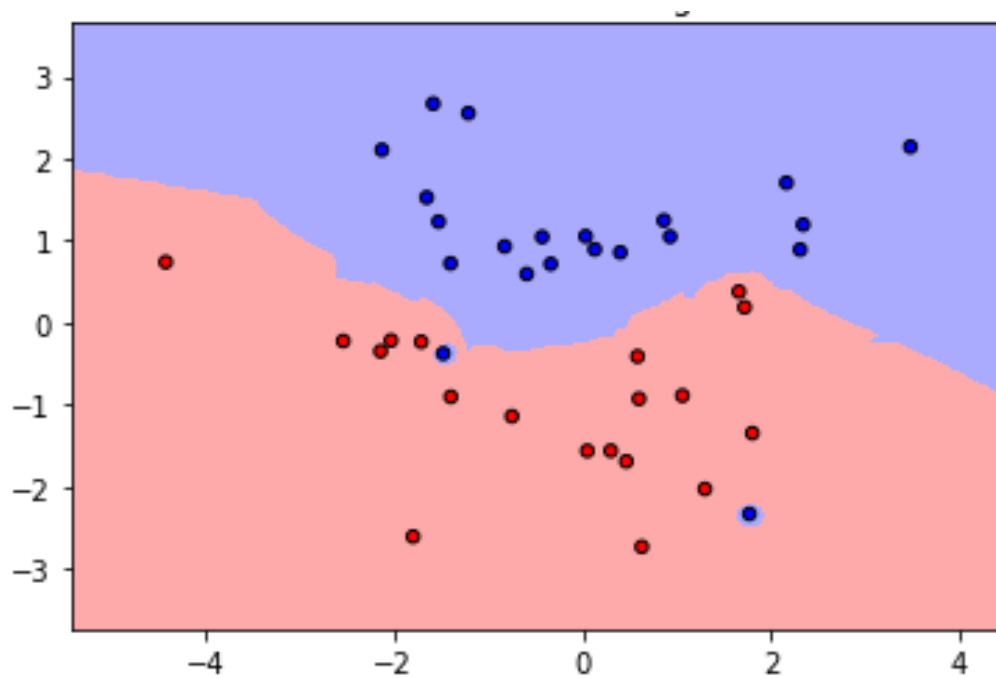
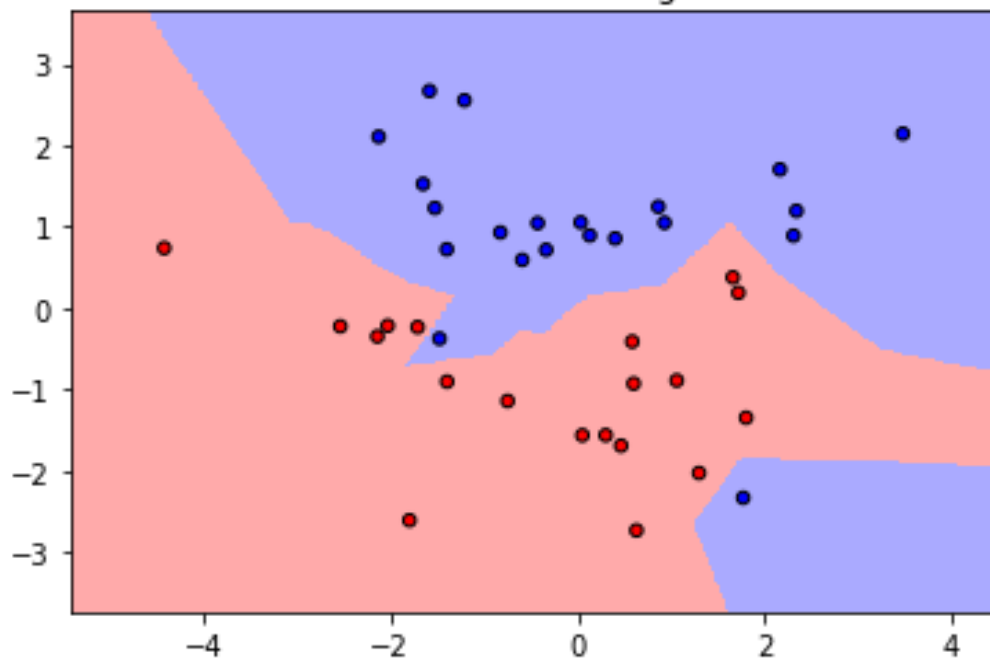
# Selecting an Appropriate K Value

- ▶ The selection of an appropriate value of  $k$  is very important.
- ▶ Selecting **too small** a value can make the algorithm **susceptible to noise** and can **overfit**.
- ▶ Selecting larger values of  $k$  lessen the impact of noise on the classification but make **boundaries between classes less distinct**.
- ▶ There are many techniques for selecting a  $k$  value.
  - ▶ Certain rules-of-thumb such as use the square root of the number of classified instances [not recommended].
  - ▶ Instead you should select a range of different  $k$  values and assess the performance of your model for these values (later when we cover scikit learn we will look at using N-fold cross validation and search to identify good values for  $k$ )









# k - Nearest Neighbour Distance Metric

- ▶ There are a number of distance measurements used to determine similarity:
  - ▶ Euclidean Distance
  - ▶ Manhattan Distance
  - ▶ Minkowski
  - ▶ Hamming Distance

# Distance Metrics

- ▶ An important aspect of k-NN algorithms is how we determine which instances are the nearest to the target case. Thus, the distance metric is a measure of the similarity between two cases.
- ▶ Common distance metrics include:
  - ▶ Euclidean
  - ▶ Manhattan
  - ▶ Minkowski
- ▶ To help illustrate the various metrics let's assume we have the dataset below with n features and two instances p and q

	Feature 1	Features 2	....	Feature n
p	$p_1$	$p_2$	.....	$p_n$
q	$q_1$	$q_2$	.....	$q_n$

# Euclidean Distance Metric

- ▶ The most common measure of distance is **Euclidian distance**: which measures the straight-line distance between two points.
- ▶ If  $\mathbf{p} = \langle p_1, p_2, \dots, p_n \rangle$  and  $\mathbf{q} = \langle q_1, q_2, \dots, q_n \rangle$  are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by

$$\begin{aligned} d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \end{aligned}$$

It is important to understand that p and q here represent two data instances. Each instance consisting of a finite set of features. The instance q has the features q1, q2, ... qn.



# Euclidean Distance Metric

Movie title	# of clicks	# of classes	Movie title	Distance to movie “?”
California Man	3	104	California Man	20.5
He’s Not Really into Dudes	2	100	He’s Not Really into Dudes	18.7
Beautiful Woman	1	81	Beautiful Woman	19.2
Kevin Longblade	101	10	Kevin Longblade	115.3
Robo Slayer 3000	99	5	Robo Slayer 3000	117.4
Amped II	98	2	Amped II	118.9
?	18	90	Action	
			Unknown	

Distance between **California man** (3, 104) and the **query** instance (18, 90) would be:

# Euclidean Distance Metric

Movie title	# of clicks	# of classes	Movie title	Distance to movie “?”
California Man	3	104	California Man	20.5
He’s Not Really into Dudes	2	100	He’s Not Really into Dudes	18.7
Beautiful Woman	1	81	Beautiful Woman	19.2
Kevin Longblade	101	10	Kevin Longblade	115.3
Robo Slayer 3000	99	5	Robo Slayer 3000	117.4
Amped II	98	2	Amped II	118.9
?	18	90	Action	
			Unknown	

Distance between **California man** (3, 104) and the **query** instance (18, 90) would be

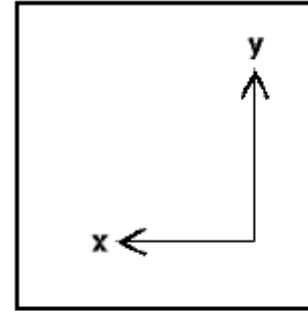
$$\sqrt{(3 - 18)^2 + (104 - 90)^2} = \sqrt{225+196}= 20.5$$

# Manhattan Distance Metric

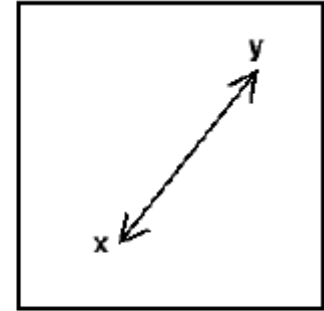
- ▶ Manhattan distance measures distance **parallel to each axis**, not diagonally (in downtown Manhattan, to get from one point to another you generally walk North-South and East-West, rather than 'as the crow flies').
- ▶ In other words take the sum of the absolute values of the differences of the coordinates
- ▶  $d(p, q) = |q_1 - p_1| + |q_2 - p_2| + |q_n - p_n| = \sum_{i=1}^n |q_i - p_i|$

# Manhattan Distance Metric

- ▶ Lets assume we have a simple dataset containing three instances as follows. We are also given the query instance below. Calculate the Manhattan and Euclidean distance between the **first training example** and the **query**



Manhattan



Euclidean

Area	Weight	Height	Capacity
10	8	4	14
12	10	6	12
14	9	4	11

Area	Weight	Height	Capacity
5	4	4	10

Area	Weight	Height	Capacity
10	8	4	8
12	10	6	12
14	9	4	11

Area	Weight	Height	Capacity
5	4	4	3

# Manhattan Distance Metric

- ▶ Euclidean

- ▶  $(10-5)^2 + (8-4)^2 + (4-4)^2 + (8-3)^2$

- ▶ 57

- ▶ Square root of 57 = 7.55

- ▶ Manhattan

- ▶  $|10-5| + |8-4| + |4-4| + |8-3|$

- ▶  $5 + 4 + 0 + 5 = 14$

Area	Weight	Height	Capacity
10	8	4	8
12	10	6	12
14	9	4	11

Area	Weight	Height	Capacity
5	4	4	3

# Minkowski Distance

- ▶ The Minkowski distance between a feature vector  $\mathbf{p} = \langle p_1, p_2, \dots, p_n \rangle$  and another feature vector  $\mathbf{q} = \langle q_1, q_2, \dots, q_n \rangle$  is defined as:
- ▶ 
$$d(p, q) = (\sum_{i=1}^n |p_i - q_i|^a)^{\frac{1}{a}}$$
- ▶ In the above equation  $a$  is an integer. Consider the case when  $a = 1$  or  $a = 2$ . Any comments.

# Minkowski Distance

- ▶ The Minkowski distance between a feature vector  $\mathbf{p} = \langle p_1, p_2, \dots, p_n \rangle$  and another feature vector  $\mathbf{q} = \langle q_1, q_2, \dots, q_n \rangle$  is defined as:
- ▶ 
$$d(p, q) = (\sum_{i=1}^n |p_i - q_i|^a)^{\frac{1}{a}}$$
- ▶ In the above equation  $a$  is an integer. Consider the case when  $a = 1$  or  $a = 2$ . Any comments.
  - ▶ For  $a = 1$  we get the Manhattan distance and for  $a = 2$  we get the Euclidean distance.
  - ▶ Minkowski Distance is a generalization of the Euclidean and Manhattan distance metrics.