# Supervised Machine Learning: Pattern Classification

# Classification

- Problem of identifying to which of a set of categories a new observation belongs
- Predicts categorical labels
- Example:
  - Predicting a person as adult or child (2-class)
  - Predicting the raise in salary based on the year of experience and salary (2-class)
  - Identify an email as spam or not (2-class)
  - Predicting the presence or absence of disease (2-class)
    - **Pima Indians Diabetes Database:** predict whether a patient has diabetes or not based on diagnostic measurements
  - Categorising the disease according to symptoms (Multi-class)
  - Categorizing the Iris flowers (Multi-class)

# Classification

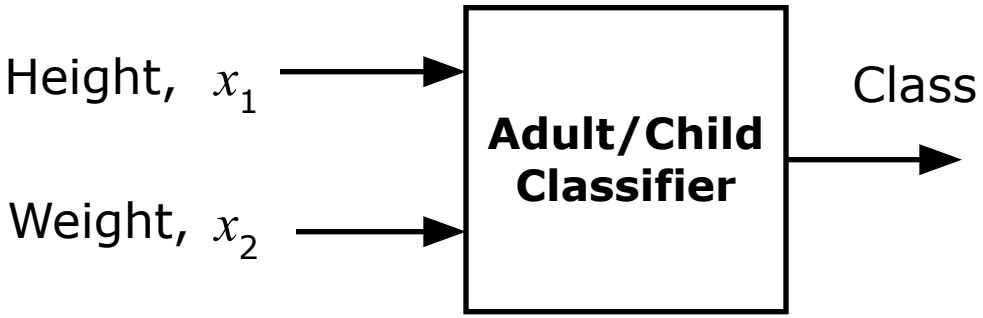
- Classification is a two step process
  - Step1: Building a classifier (data modeling)
    - Learning from data (training phase)
    - Supervised learning: In supervised learning, each example is a *pair* consisting of an input example and a desired output value (class label)
    - Training phase or learning phase is viewed as the learning of a mapping or function that can predict the associated class label of a given training example

$$y_n = f(\mathbf{x}_n)$$

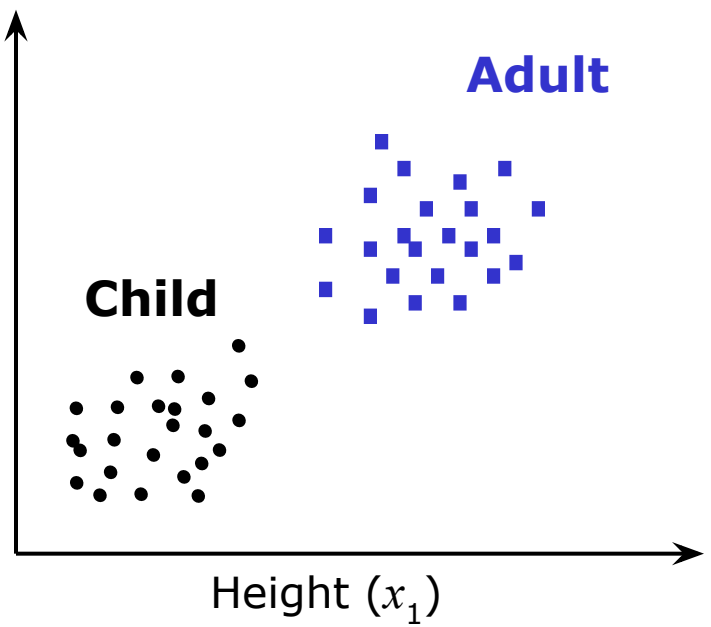
      - $\mathbf{x}_n$ is the $n^{\text{th}}$ training example and $y_n$ is the associated class label
  - Step2: Using classification model for prediction
    - Testing phase - Predicting class label for the unseen data
- Accuracy of a classifier: Percentage of test examples that are correctly classified by the classifier
- Target of learning techniques: Good generalization ability

# 2-class Classification

- Example: Classifying a person as child or adult




Adult :Class $C_1$
Child :Class $C_2$

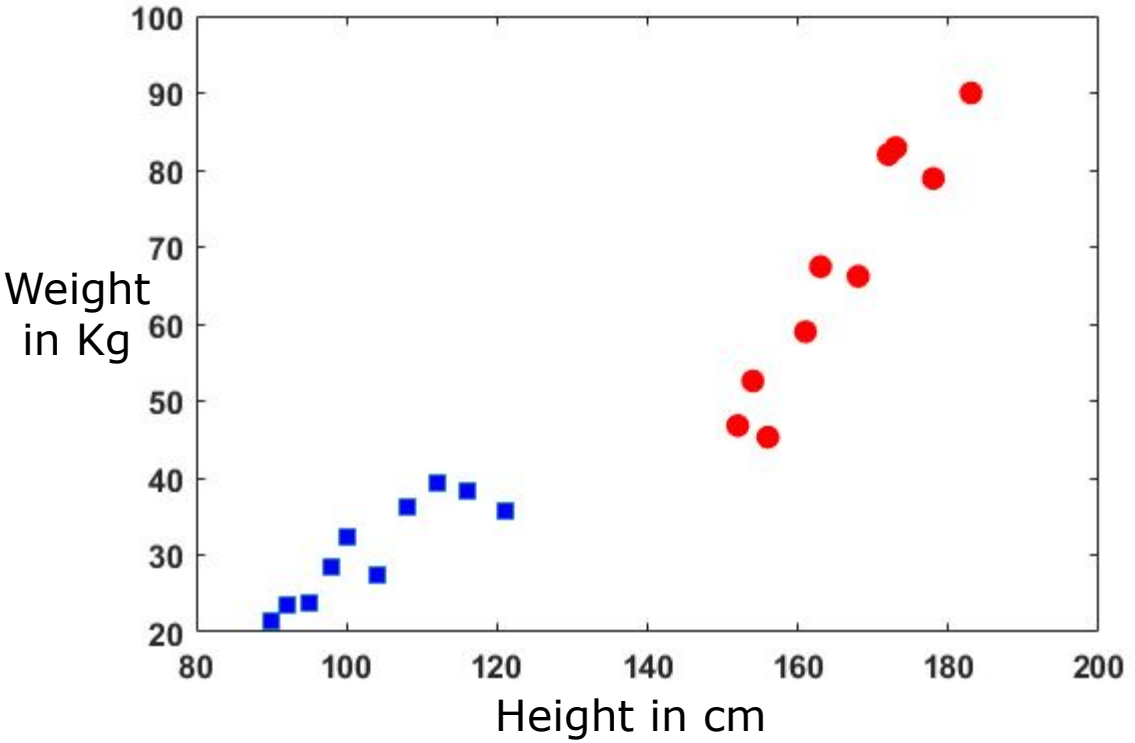



$$\mathbf{x} = [x_1 \;\; x_2]^T$$

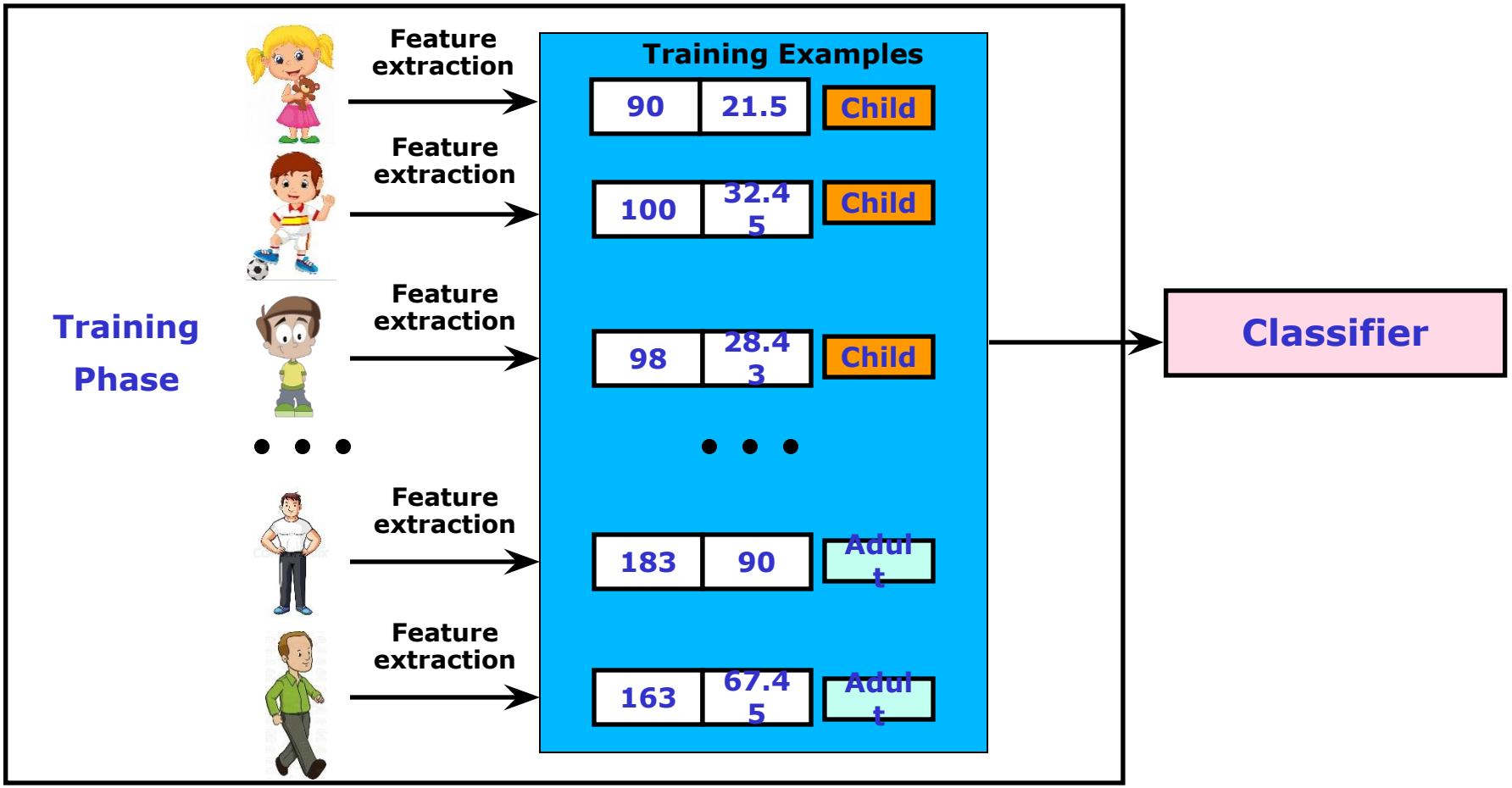
# Illustration of Training Set: Adult-Child

| Height | Weight | Class |
|---|---|---|
| 90 | 21.5 | 0 |
| 95 | 23.67 | 0 |
| 100 | 32.45 | 0 |
| 116 | 38.21 | 0 |
| 98 | 28.43 | 0 |
| 108 | 36.32 | 0 |
| 104 | 27.38 | 0 |
| 112 | 39.28 | 0 |
| 121 | 35.8 | 0 |
| 92 | 23.56 | 0 |
| 152 | 46.8 | 1 |
| 178 | 78.9 | 1 |
| 163 | 67.45 | 1 |
| 173 | 82.9 | 1 |
| 154 | 52.6 | 1 |
| 168 | 66.2 | 1 |
| 183 | 90 | 1 |
| 172 | 82 | 1 |
| 156 | 45.3 | 1 |
| 161 | 59 | 1 |

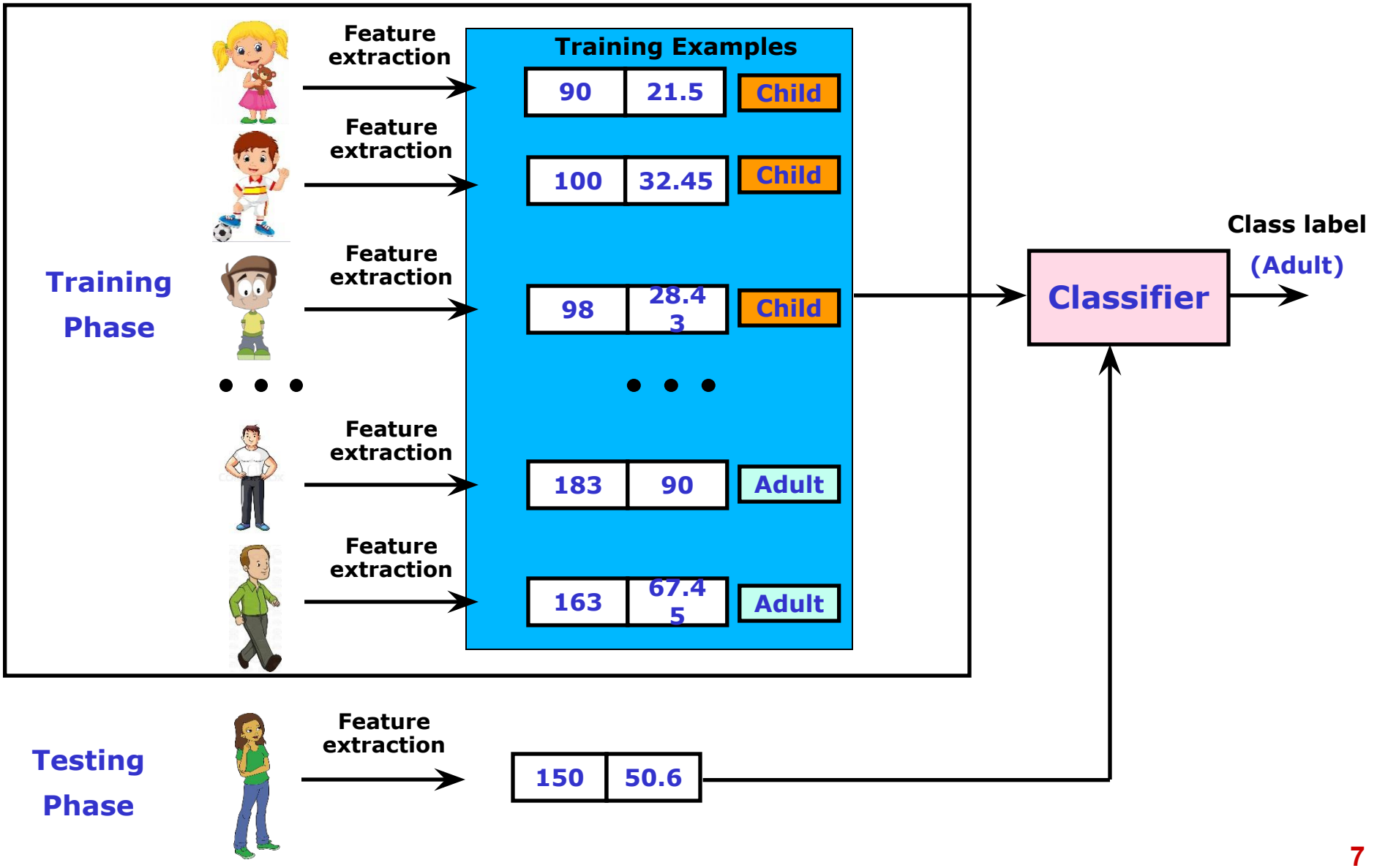
- Number of training examples ($N$) = 20
- Dimension of a training example = 2
- Class label attribute is 3rd dimension
- Class:
  - Child (0)
  - Adult (1)



# Step1: Building a Classification Model (Training Phase)



# Step2: Classification (Testing Phase)



# Data Preparation for the Classification

- Divide the data into training set and test set
- **Approach 1:** When the number samples from each class are almost equal (Balanced data)
  - Most common split is 70-30 split:
    - Training data contain 70% of samples from each class
    - Test data contain remaining 30% of samples from each class
  - One can use other splits like 50-50 or 60-40 or 80-20 or 90-10

# Data Preparation for the Classification: Approach 1

- Suppose that we are doing 70-30 split
- Suppose the data set has 3000 samples
- Each sample is belonging to one of the 3 classes
- Suppose each class has 1000 samples
  - Step1: From class1, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step2: From class2, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step3: From class3, 70% i.e. 700 samples considered as training samples and remaining 30% i.e. 300 samples are considered as test samples
  - Step4: Combine training examples from each class
    - Training set now contain 700+700+700=2100 samples
  - Step5: Combine test examples from each class
    - Test set now contain 300+300+300=900 samples

# Data Preparation for the Classification

- Divide the data into training set and test set
- **Approach 1:** When the number samples from each class are almost equal (Balanced data)
  - Example:
    - Training data contain 70% of samples from each class
    - Test data contain remaining 30% of samples from each class
- **Approach 2:** When the number samples from each class are not equal (Imbalanced data)
  - One class may have large number of samples and another has small number of samples
  - 70%-30% division may cause learned model to be bias to class with larger number of training samples
  - Solution:
    - Consider 70% or 80% of the samples from the class with least number of samples as training data from that class
    - Consider the same number of samples from other class as training examples
    - Each class will have same number of training examples

# Data Preparation for the Classification: Approach 2

- Suppose the data set has 3000 samples
- Each sample is belonging to one of the 3 classes
- Suppose class1 has 700 samples, class2 has 300 samples and class3 has 2000 samples
  - Step1: From class2, 70% i.e. 210 samples considered as training samples and remaining 30% i.e. 90 samples are considered as test samples
  - Step2: From class1, 210 samples considered as training samples and remaining 490 samples are considered as test samples
  - Step3: From class3, 210 samples considered as training samples and remaining 1790 samples are considered as test samples
  - Step4: Combine training examples from each class
    - Training set now contain 210+210+210=630 samples
  - Step5: Combine test examples from each class
    - Test set now contain 490+90+1790=2370 samples

# Nearest-Neighbour Method

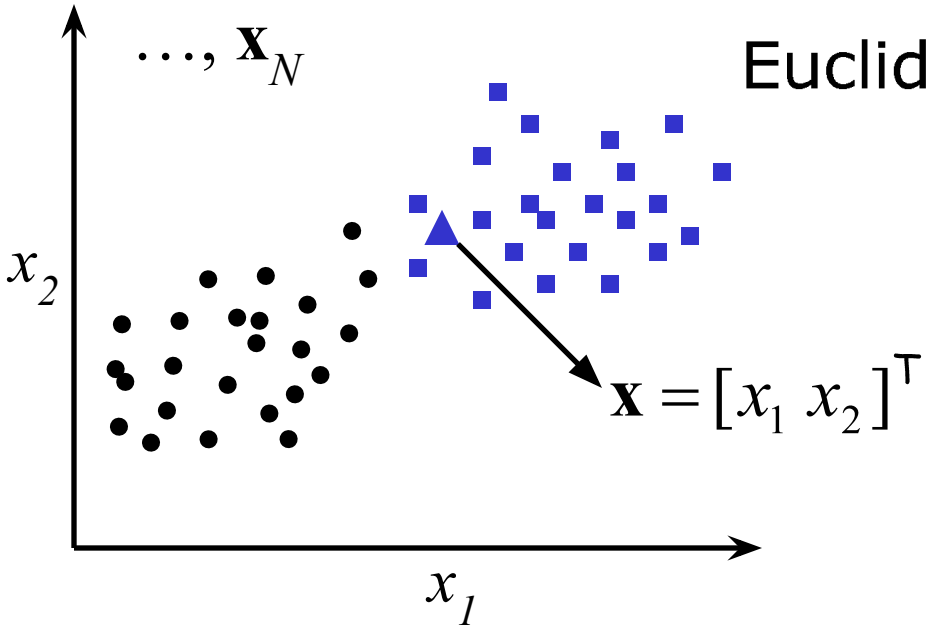
- Training data with $N$ samples: $\mathrm{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

  $\mathbf{x}_n \in \mathrm{R}^d$ and $y_n \in \{1, 2, \ldots, M\}$

  – $d$: dimension of input example

  – $M$: Number of classes

- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$




$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$

$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^\mathsf{T}(\mathbf{x}_n - \mathbf{x})}$$

$$= \sqrt{\sum_{i=1}^{d}(x_{ni} - x_i)^2}$$

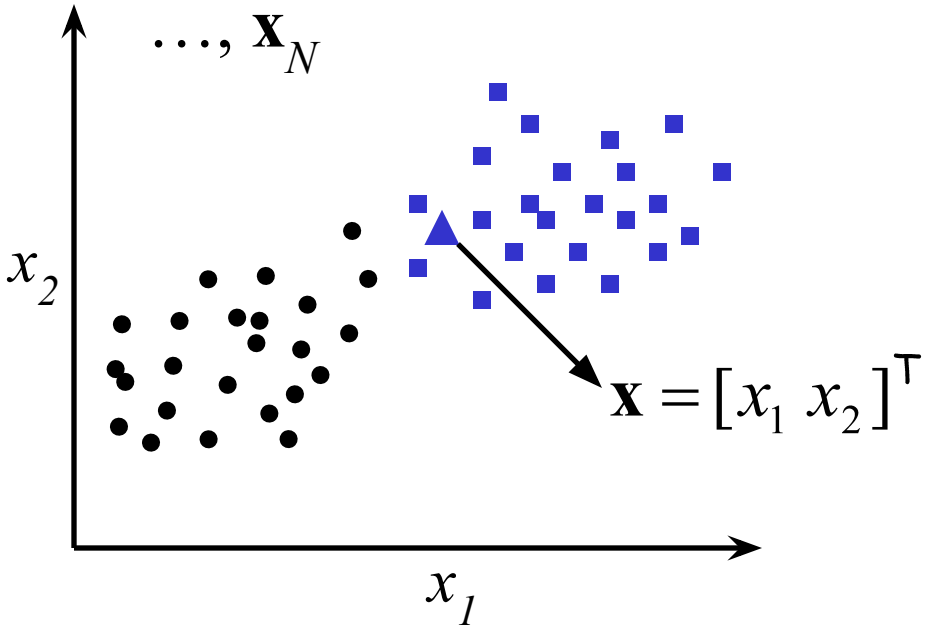
# Nearest-Neighbour Method

- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

$$\mathbf{x}_n \in \mathrm{R}^{d} \text{ and } y_n \in \{1, 2, \ldots, M\}$$

  - $d$: dimension of input example
  - $M$: Number of classes

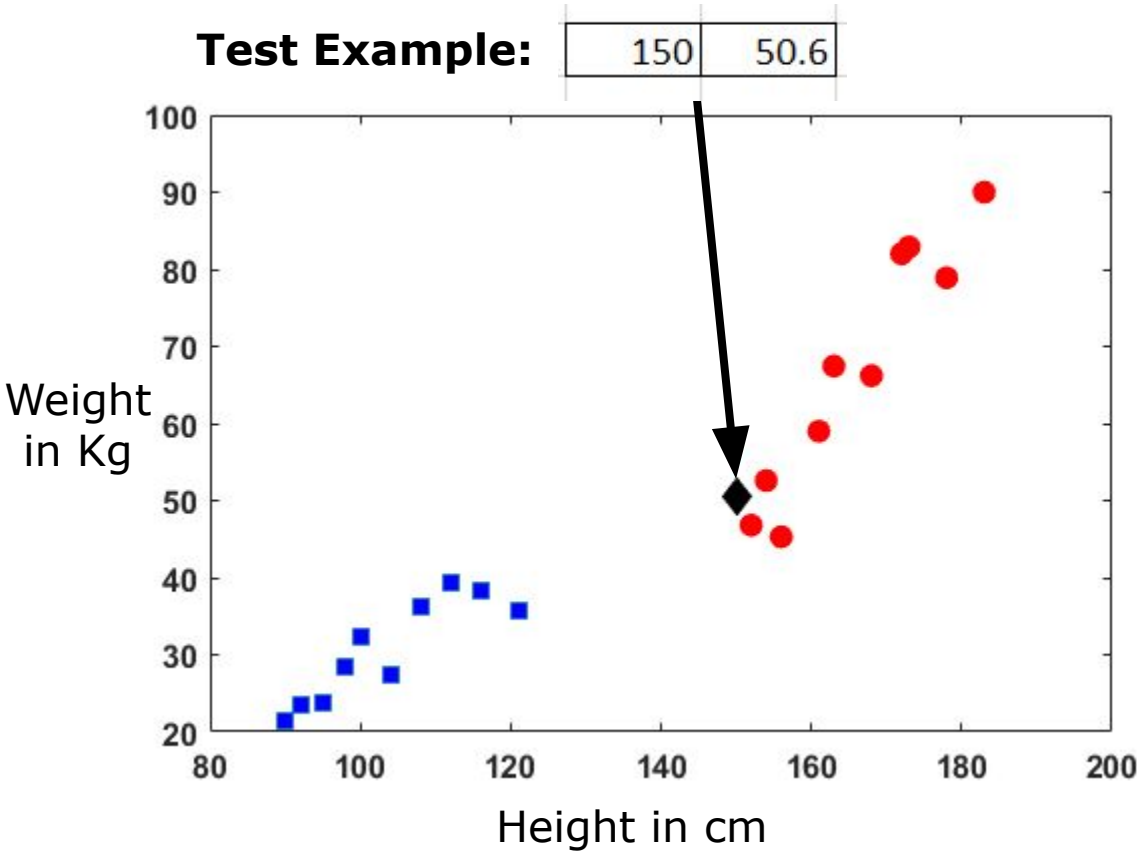
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$




- Step 2: Sort the examples in the training set in the ascending order of the distance to test example $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

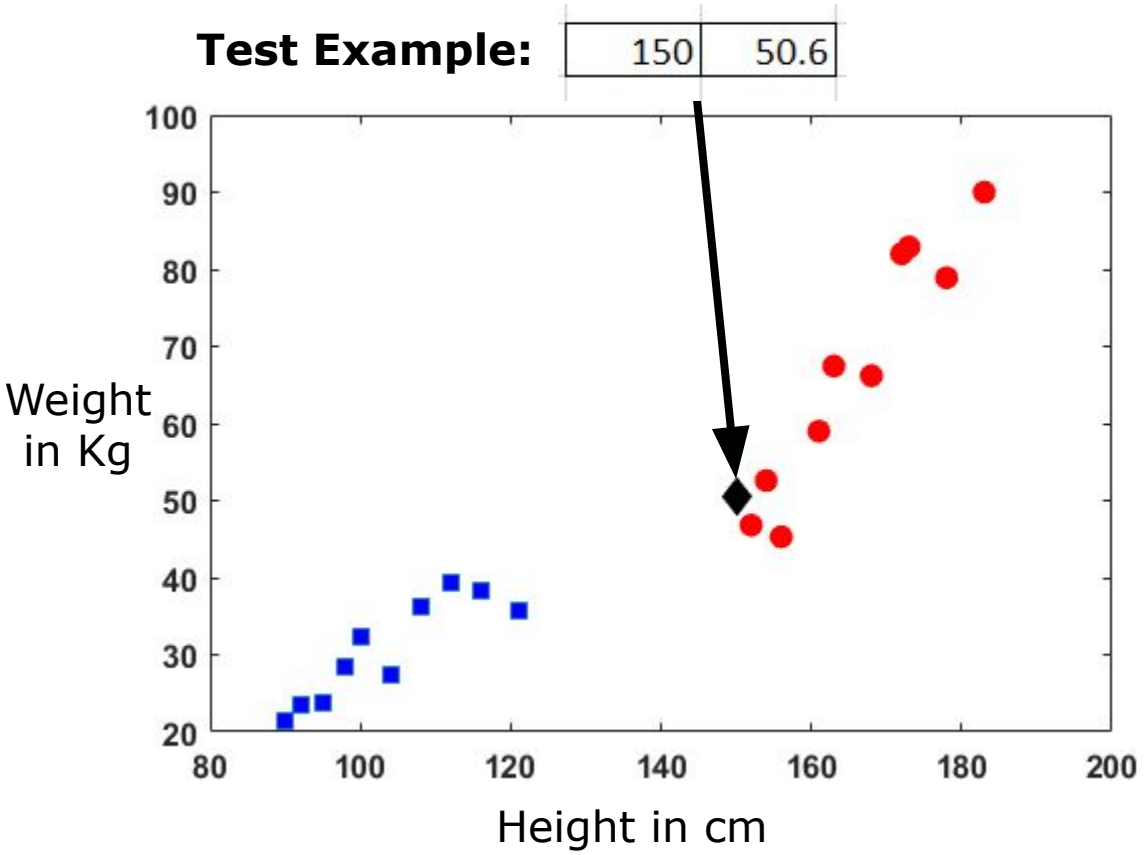
| Height | Weight | ED |
|---|---|---|
| 90 | 21.5 | 66.68 |
| 95 | 23.67 | 61.24 |
| 100 | 32.45 | 53.19 |
| 116 | 38.21 | 36.19 |
| 98 | 28.43 | 56.53 |
| 108 | 36.32 | 44.36 |
| 104 | 27.38 | 51.53 |
| 112 | 39.28 | 39.65 |
| 121 | 35.8 | 32.56 |
| 92 | 23.56 | 63.99 |
| 152 | 46.8 | 4.294 |
| 178 | 78.9 | 39.81 |
| 163 | 67.45 | 21.28 |
| 173 | 82.9 | 39.65 |
| 154 | 52.6 | 4.472 |
| 168 | 66.2 | 23.82 |
| 183 | 90 | 51.39 |
| 172 | 82 | 38.34 |
| 156 | 45.3 | 8.006 |
| 161 | 59 | 13.84 |




- Step 1: Compute Euclidean distance (ED) will each training examples

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

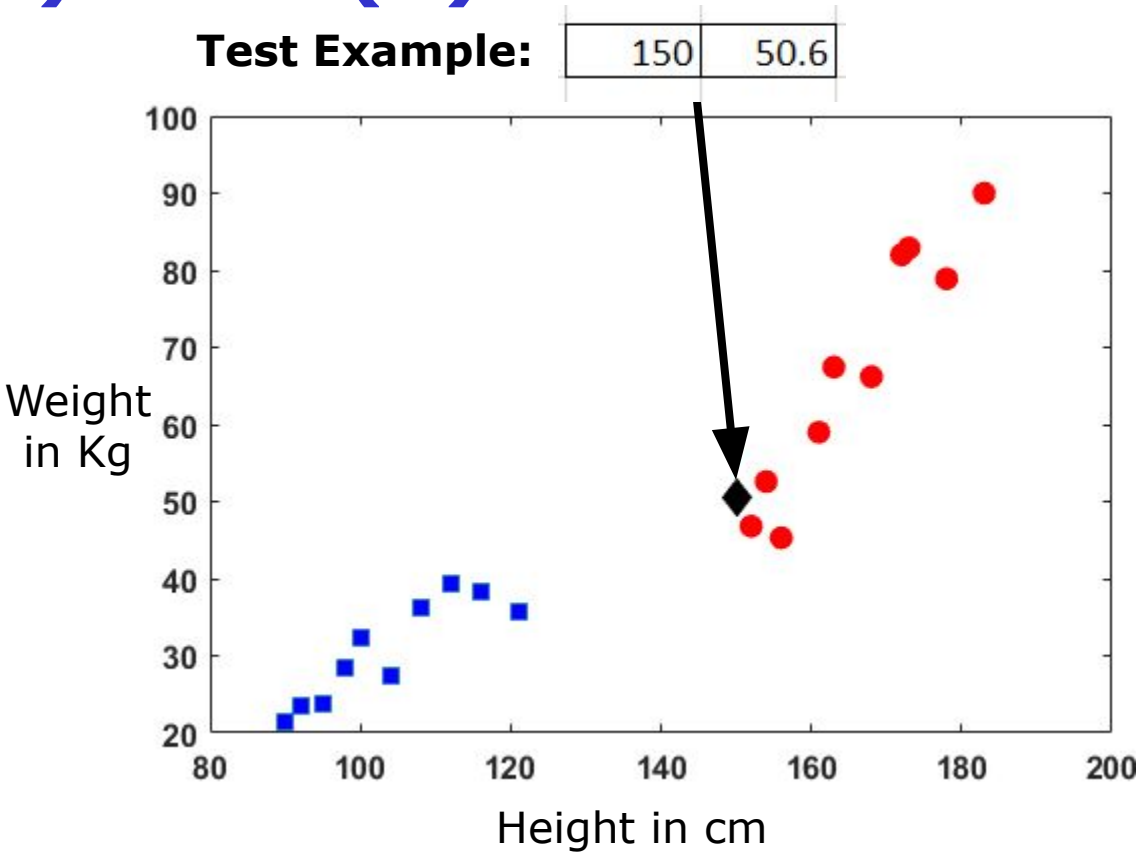
| Height | Weight | ED |
|---|---|---|
| 90 | 21.5 | 66.68 |
| 95 | 23.67 | 61.24 |
| 100 | 32.45 | 53.19 |
| 116 | 38.21 | 36.19 |
| 98 | 28.43 | 56.53 |
| 108 | 36.32 | 44.36 |
| 104 | 27.38 | 51.53 |
| 112 | 39.28 | 39.65 |
| 121 | 35.8 | 32.56 |
| 92 | 23.56 | 63.99 |
| 152 | 46.8 | 4.294 |
| 178 | 78.9 | 39.81 |
| 163 | 67.45 | 21.28 |
| 173 | 82.9 | 39.65 |
| 154 | 52.6 | 4.472 |
| 168 | 66.2 | 23.82 |
| 183 | 90 | 51.39 |
| 172 | 82 | 38.34 |
| 156 | 45.3 | 8.006 |
| 161 | 59 | 13.84 |




- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

| Class | Height | Weight | ED |
|---|---|---|---|
| 0 | 90 | 21.5 | 66.68 |
| 0 | 95 | 23.67 | 61.24 |
| 0 | 100 | 32.45 | 53.19 |
| 0 | 116 | 38.21 | 36.19 |
| 0 | 98 | 28.43 | 56.53 |
| 0 | 108 | 36.32 | 44.36 |
| 0 | 104 | 27.38 | 51.53 |
| 0 | 112 | 39.28 | 39.65 |
| 0 | 121 | 35.8 | 32.56 |
| 0 | 92 | 23.56 | 63.99 |
| 1 | 152 | 46.8 | 4.294 |
| 1 | 178 | 78.9 | 39.81 |
| 1 | 163 | 67.45 | 21.28 |
| 1 | 173 | 82.9 | 39.65 |
| 1 | 154 | 52.6 | 4.472 |
| 1 | 168 | 66.2 | 23.82 |
| 1 | 183 | 90 | 51.39 |
| 1 | 172 | 82 | 38.34 |
| 1 | 156 | 45.3 | 8.006 |
| 1 | 161 | 59 | 13.84 |




- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult (1)

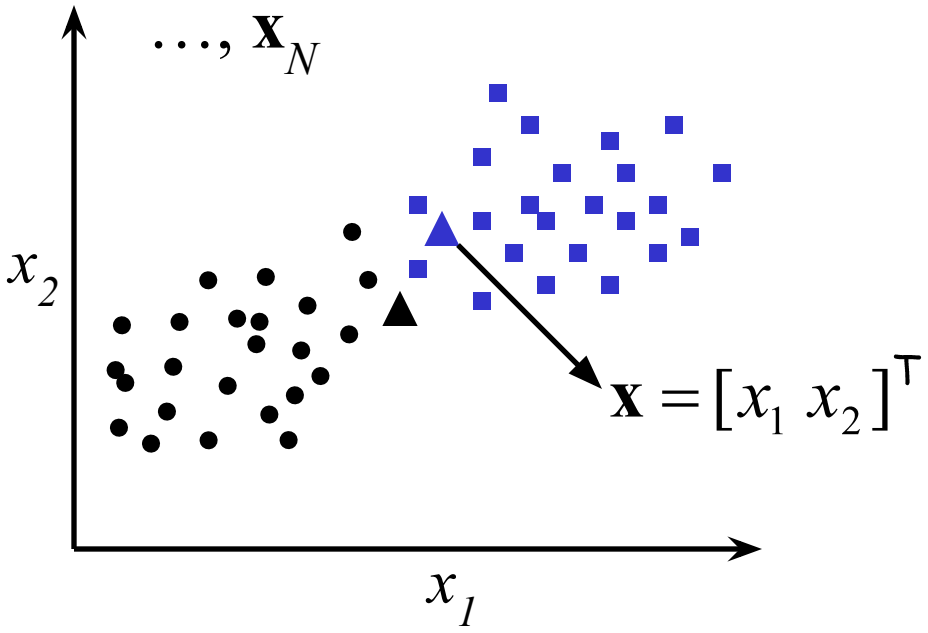
# Nearest-Neighbour Method

- Training data: $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$,

$$\mathbf{x}_n \in \mathrm{R}^d \text{ and } y_n \in \{1, 2, \ldots, M\}$$

  - $d$: dimension of input example
  - $M$: Number of classes

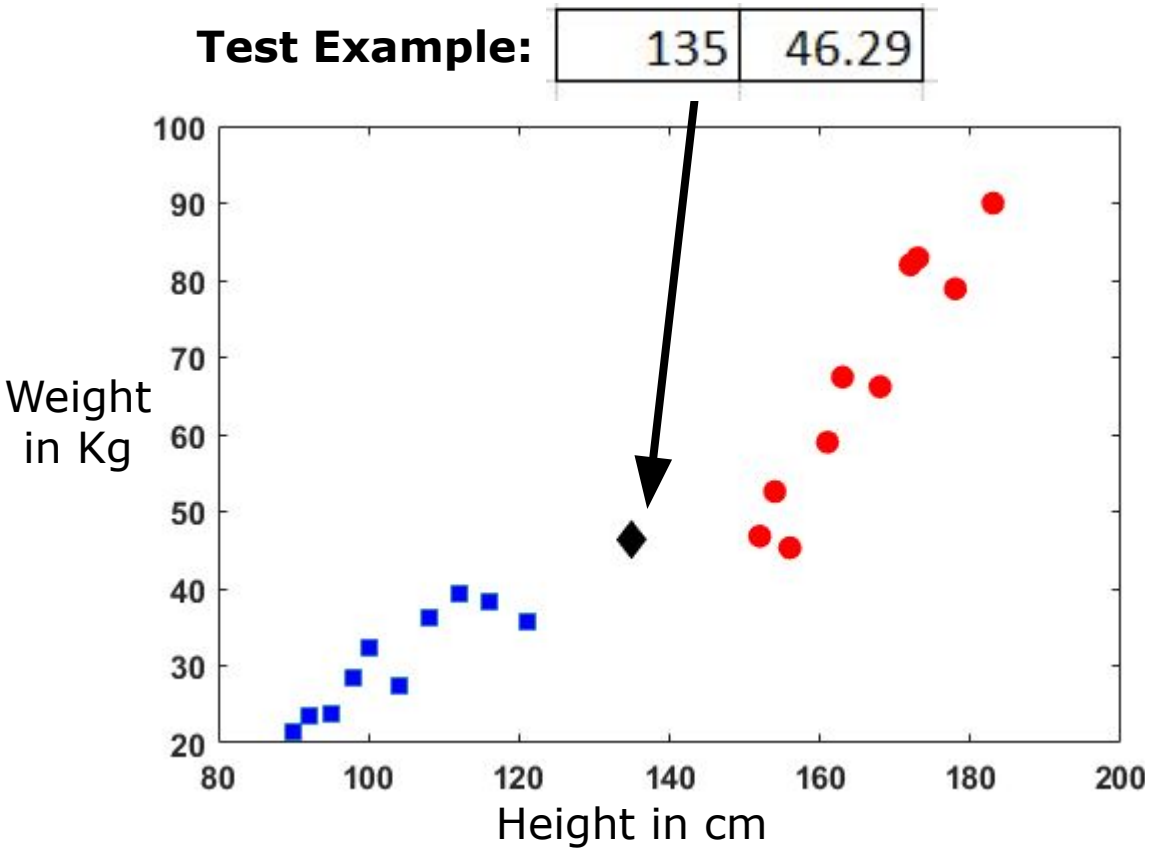
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$




- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Assign the class of the training example with the minimum distance to the test example, $\mathbf{x}$

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

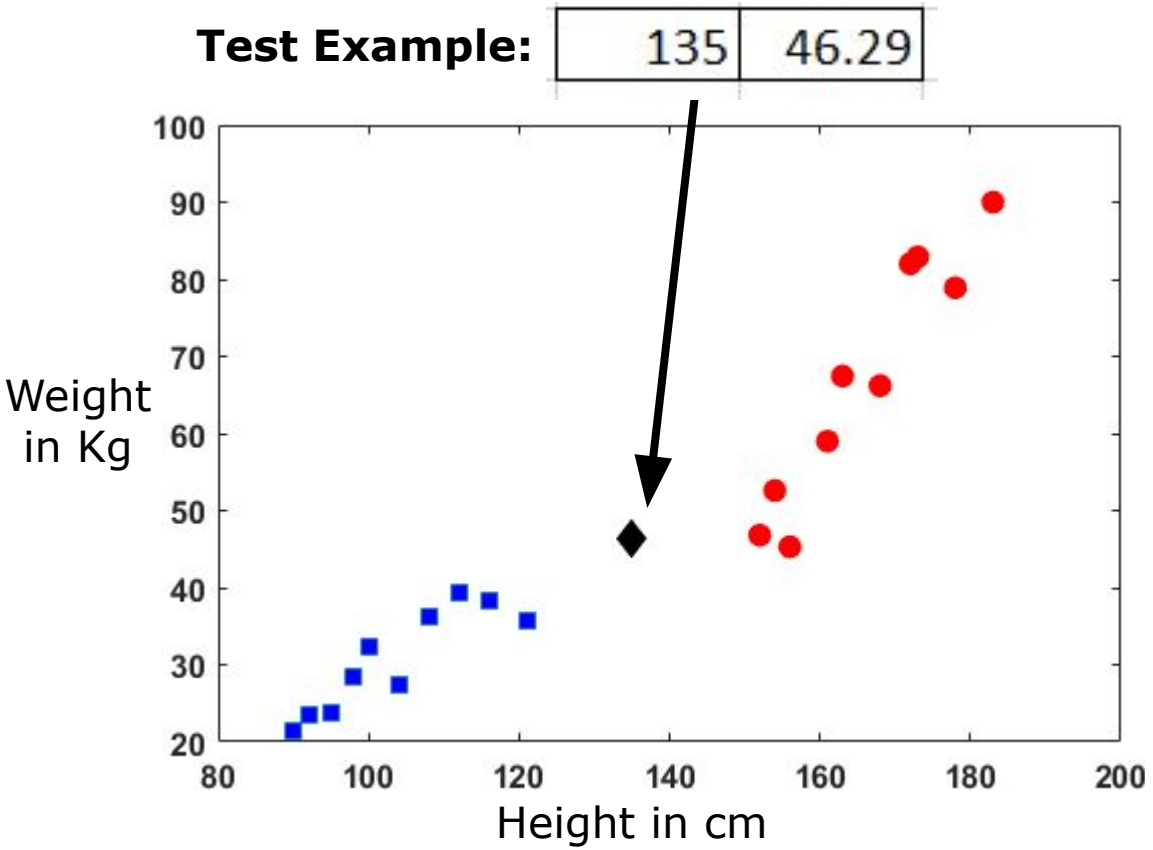
| Height | Weight | ED |
|---|---|---|
| 90 | 21.5 | 51.38 |
| 95 | 23.67 | 45.95 |
| 100 | 32.45 | 37.64 |
| 116 | 38.21 | 20.65 |
| 98 | 28.43 | 41.09 |
| 108 | 36.32 | 28.78 |
| 104 | 27.38 | 36.31 |
| 112 | 39.28 | 24.04 |
| 121 | 35.8 | 17.49 |
| 92 | 23.56 | 48.64 |
| 152 | 46.8 | 17.01 |
| 178 | 78.9 | 53.97 |
| 163 | 67.45 | 35.1 |
| 173 | 82.9 | 52.77 |
| 154 | 52.6 | 20.02 |
| 168 | 66.2 | 38.54 |
| 183 | 90 | 64.92 |
| 172 | 82 | 51.42 |
| 156 | 45.3 | 21.02 |
| 161 | 59 | 28.94 |




- Step 1: Compute Euclidean distance (ED) will each training examples

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

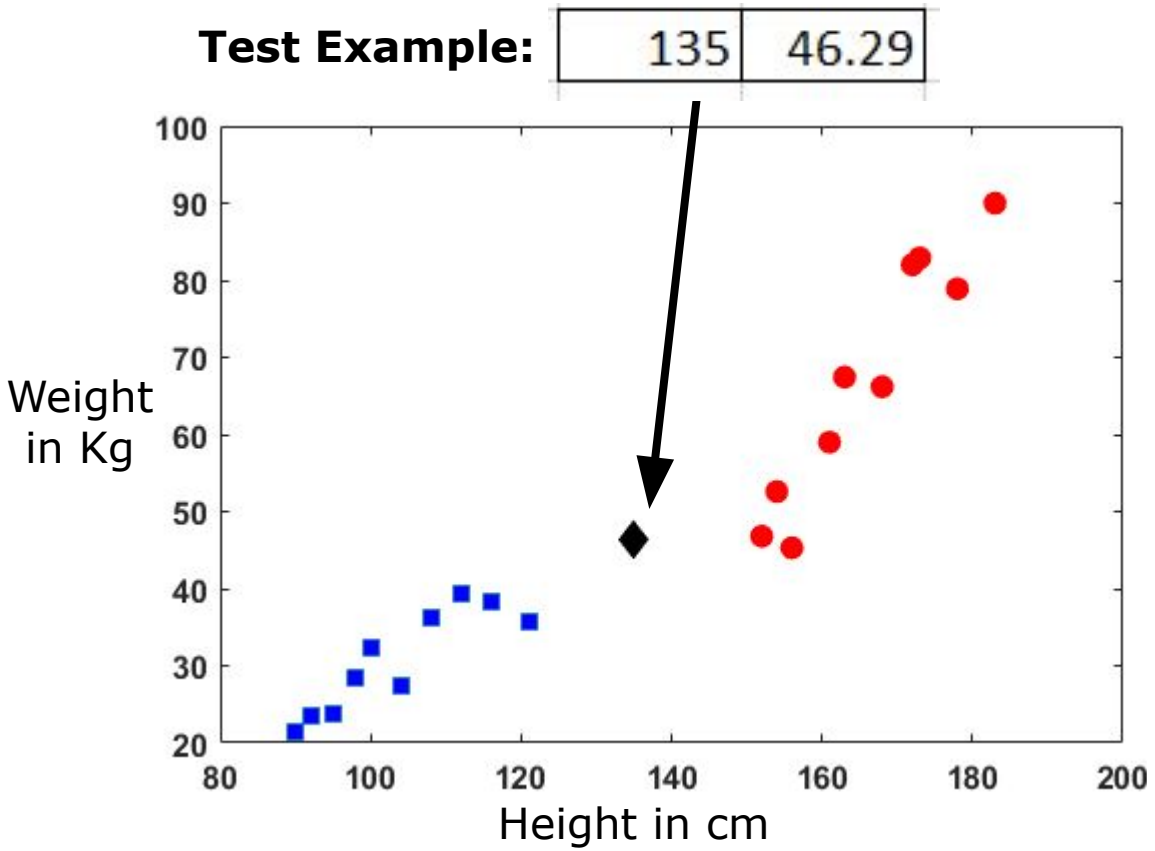
| Height | Weight | ED |
|---|---|---|
| 90 | 21.5 | 51.38 |
| 95 | 23.67 | 45.95 |
| 100 | 32.45 | 37.64 |
| 116 | 38.21 | 20.65 |
| 98 | 28.43 | 41.09 |
| 108 | 36.32 | 28.78 |
| 104 | 27.38 | 36.31 |
| 112 | 39.28 | 24.04 |
| 121 | 35.8 | 17.49 |
| 92 | 23.56 | 48.64 |
| 152 | 46.8 | 17.01 |
| 178 | 78.9 | 53.97 |
| 163 | 67.45 | 35.1 |
| 173 | 82.9 | 52.77 |
| 154 | 52.6 | 20.02 |
| 168 | 66.2 | 38.54 |
| 183 | 90 | 64.92 |
| 172 | 82 | 51.42 |
| 156 | 45.3 | 21.02 |
| 161 | 59 | 28.94 |




- Step 2: Sort the examples in the training set in the ascending order of the distance to test example

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

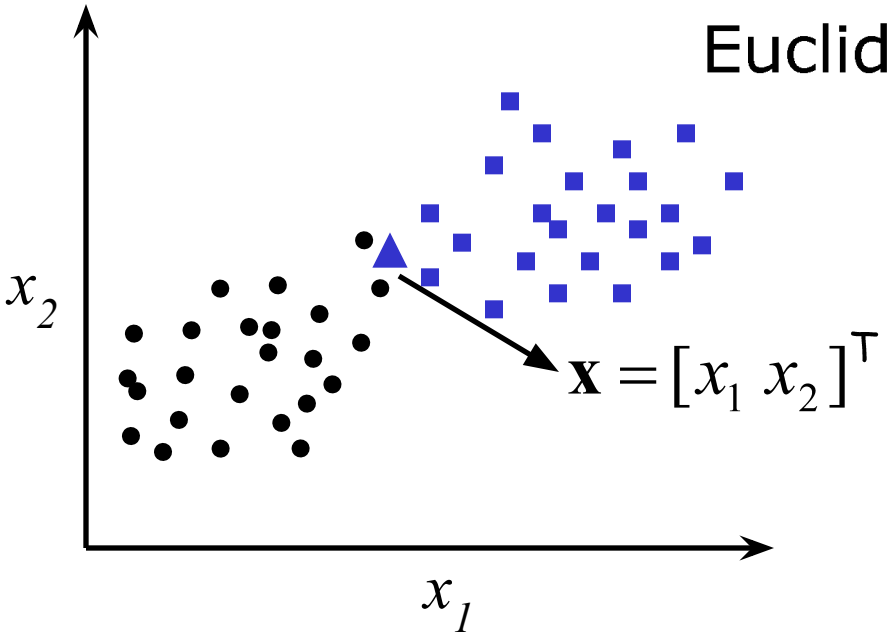
| Class | Height | Weight | ED |
|---|---|---|---|
| 0 | 90 | 21.5 | 51.38 |
| 0 | 95 | 23.67 | 45.95 |
| 0 | 100 | 32.45 | 37.64 |
| 0 | 116 | 38.21 | 20.65 |
| 0 | 98 | 28.43 | 41.09 |
| 0 | 108 | 36.32 | 28.78 |
| 0 | 104 | 27.38 | 36.31 |
| 0 | 112 | 39.28 | 24.04 |
| 0 | 121 | 35.8 | 17.49 |
| 0 | 92 | 23.56 | 48.64 |
| 1 | 152 | 46.8 | 17.01 |
| 1 | 178 | 78.9 | 53.97 |
| 1 | 163 | 67.45 | 35.1 |
| 1 | 173 | 82.9 | 52.77 |
| 1 | 154 | 52.6 | 20.02 |
| 1 | 168 | 66.2 | 38.54 |
| 1 | 183 | 90 | 64.92 |
| 1 | 172 | 82 | 51.42 |
| 1 | 156 | 45.3 | 21.02 |
| 1 | 161 | 59 | 28.94 |




- Step 3: Assign the class of the training example with the minimum distance to the test example
  - Class: Adult (1) ?

# K-Nearest Neighbours (K-NN) Method

- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$



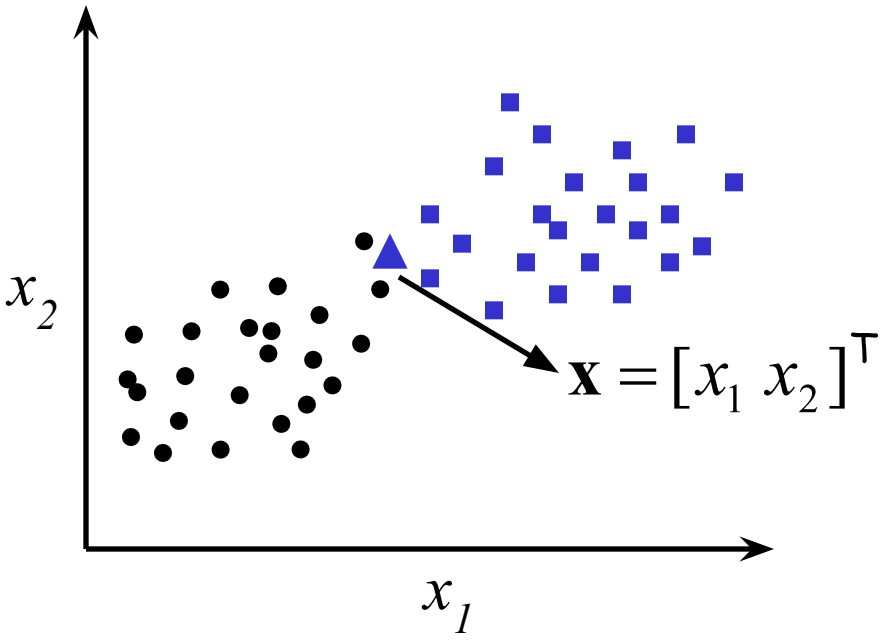

$$\text{Euclidean distance} = \|\mathbf{x}_n - \mathbf{x}\|$$

$$= \sqrt{(\mathbf{x}_n - \mathbf{x})^T (\mathbf{x}_n - \mathbf{x})}$$

$$= \sqrt{\sum_{i=1}^{d} (x_{ni} - x_i)^2}$$

# K-Nearest Neighbours (K-NN) Method

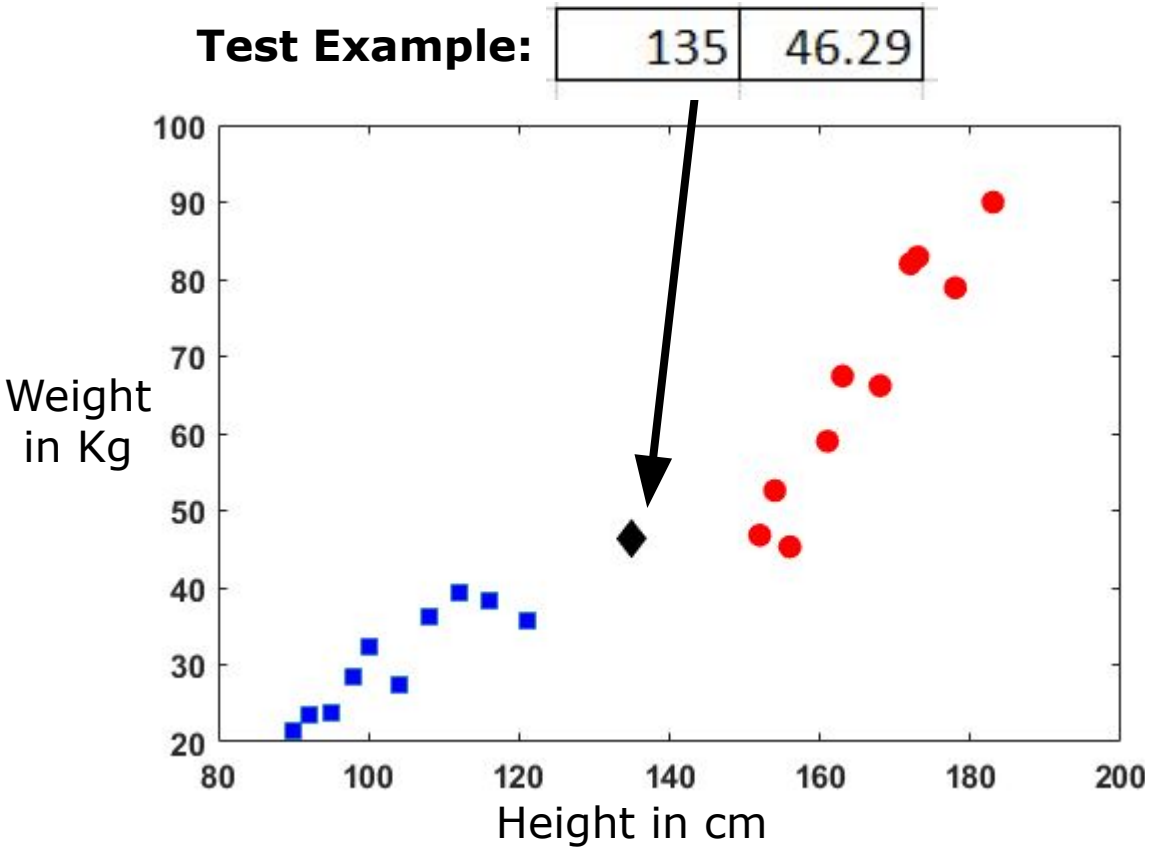
- Consider the class labels of the K training examples nearest to the test example
- Step 1: Compute Euclidean distance for a test example $\mathbf{x}$ with every training examples, $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, …, $\mathbf{x}_N$




- Step 2: Sort the examples in the training set in the ascending order of the distance to $\mathbf{x}$
- Step 3: Choose the first K examples in the sorted list
  - K is the number of neighbours for text example
- Step 4: Test example is assigned the most common class among its K neighbours

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

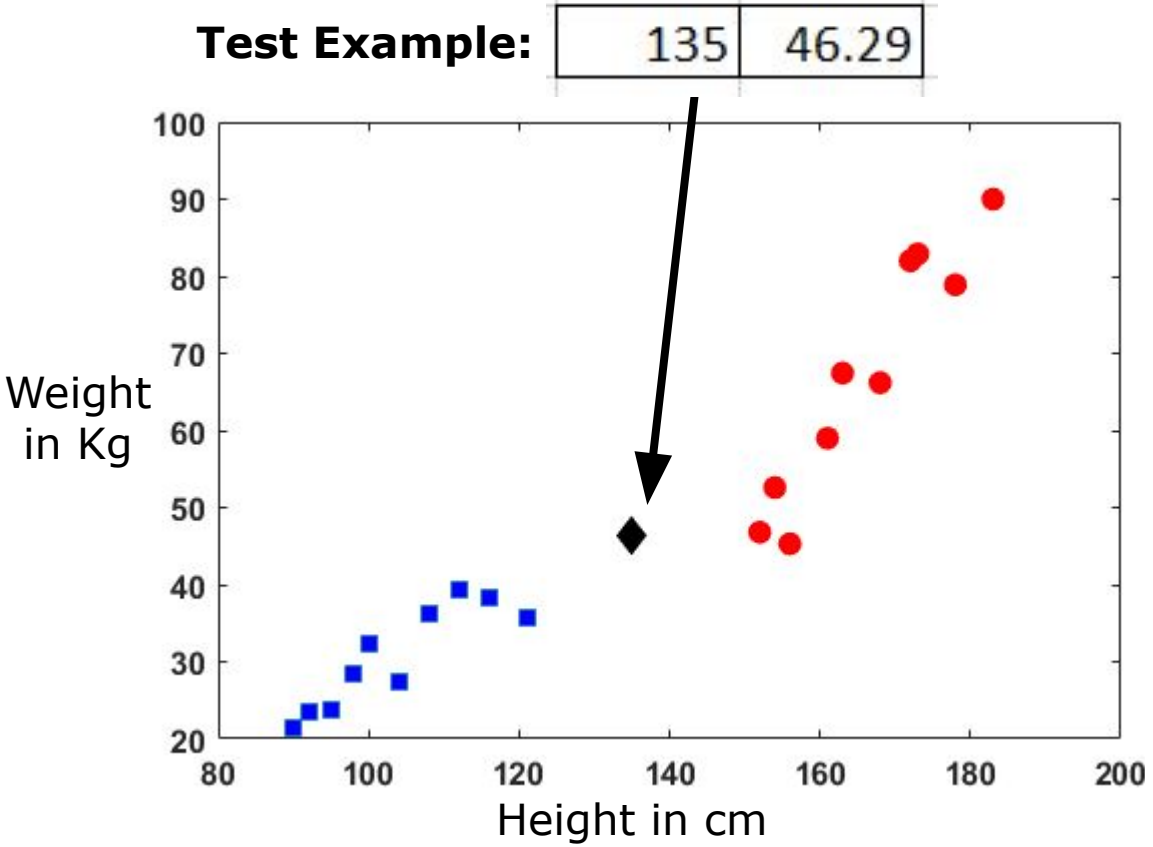
| Class | Height | Weight | ED |
|---|---|---|---|
| 0 | 90 | 21.5 | 51.38 |
| 0 | 95 | 23.67 | 45.95 |
| 0 | 100 | 32.45 | 37.64 |
| 0 | 116 | 38.21 | 20.65 |
| 0 | 98 | 28.43 | 41.09 |
| 0 | 108 | 36.32 | 28.78 |
| 0 | 104 | 27.38 | 36.31 |
| 0 | 112 | 39.28 | 24.04 |
| 0 | 121 | 35.8 | 17.49 |
| 0 | 92 | 23.56 | 48.64 |
| 1 | 152 | 46.8 | 17.01 |
| 1 | 178 | 78.9 | 53.97 |
| 1 | 163 | 67.45 | 35.1 |
| 1 | 173 | 82.9 | 52.77 |
| 1 | 154 | 52.6 | 20.02 |
| 1 | 168 | 66.2 | 38.54 |
| 1 | 183 | 90 | 64.92 |
| 1 | 172 | 82 | 51.42 |
| 1 | 156 | 45.3 | 21.02 |
| 1 | 161 | 59 | 28.94 |




- Consider K=5
- Step 3: Choose the first K=5 examples in the sorted list

# Illustration of Nearest Neighbour Method: Adult(1)-Child(0) Classification

| Class | Height | Weight | ED |
|---|---|---|---|
| 0 | 90 | 21.5 | 51.38 |
| 0 | 95 | 23.67 | 45.95 |
| 0 | 100 | 32.45 | 37.64 |
| 0 | 116 | 38.21 | 20.65 |
| 0 | 98 | 28.43 | 41.09 |
| 0 | 108 | 36.32 | 28.78 |
| 0 | 104 | 27.38 | 36.31 |
| 0 | 112 | 39.28 | 24.04 |
| 0 | 121 | 35.8 | 17.49 |
| 0 | 92 | 23.56 | 48.64 |
| 1 | 152 | 46.8 | 17.01 |
| 1 | 178 | 78.9 | 53.97 |
| 1 | 163 | 67.45 | 35.1 |
| 1 | 173 | 82.9 | 52.77 |
| 1 | 154 | 52.6 | 20.02 |
| 1 | 168 | 66.2 | 38.54 |
| 1 | 183 | 90 | 64.92 |
| 1 | 172 | 82 | 51.42 |
| 1 | 156 | 45.3 | 21.02 |
| 1 | 161 | 59 | 28.94 |




- Consider K=5
- Step 4: Test example is assigned the most common class among its K neighbours
  - Class: Adult

# Determining K, Number of Neighbours

- This is determined experimentally
- Starting with K=1, test set is used to estimate the accuracy of the classifier
- This process is repeated each time by incrementing K to allow for more neighbour
- The K value that gives the maximum accuracy may be selected
- Preferably the value of K should be an odd number and prime number.

# Data Normalization

- Since the distance measure is used, K-NN classifier require normalising the values of each attribute
- Normalising the training data:
  - Compute the minimum and maximum values of each of the attributes in the training data
  - Store the minimum and maximum values of each of the attributes
  - Perform the min-max normalization on training data set
- Normalizing the test data:
  - Use the stored minimum and maximum values of each of the attributes from training set to normalise the test examples
- NOTE: Ensure that test examples are not causing out-of-bound error

# Lazy Learning : Learning from Neighbours

- The K nearest neighbour classifier is an example of lazy learner
- Lazy learning waits until the last minute before doing any model construction to classify test example
- When the training examples are given, a lazy learner simply stores them and waits until it is given a test example
- When it sees the test example, then it classify based on its similarity to the stored training examples
- Since the lazy learns stores the training examples or instances, they also called instance based learners
- Disadvantages:
  - Making classification or prediction is computationally intensive
  - Require efficient huge storage techniques when the training samples are huge

# Text Books


1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.

2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.

3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.