CIS 678 Machine Learning

ML Introduction: Linear Regression (part 1)

What we'd like to accomplish today

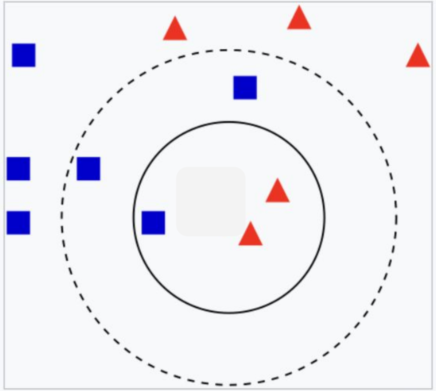
- General Concepts: ***Straight Line*** *to* ***Linear Regression***
- Gradient Descent Algorithm
    - ➜ A simple two parameter **Linear Regression** model
    - ➜ Hands on **Notebook implementation**
- QA

What we have learned so far!

- k-Nearest neighbors (k-NN)

Our first ML Model

- You are given a set of data points of two classes: red triangles, and blue squares

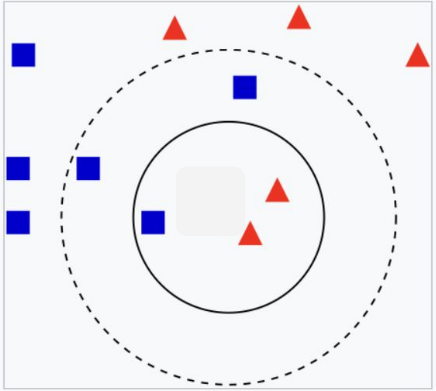


Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If $k = 3$ (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Our first ML Model

- You are given a set of data points of two classes: red triangles, and blue squares
- And asked to develop a ML model that can classify (a new data point)between these two classes.



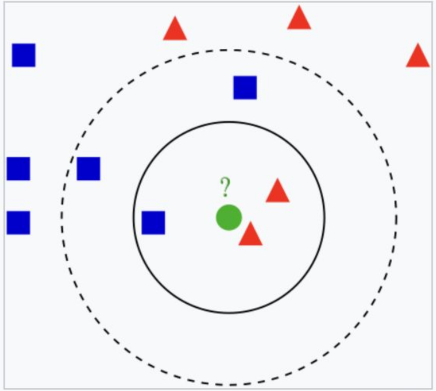
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Our first ML Model

- k-Nearest neighbors (k-NN)
  - Supervised learning
  - Non parametric ***(Distance based method)***
  - Both for Classification and Regression solutions

Circles are drawn using L2/Euclidean Distance



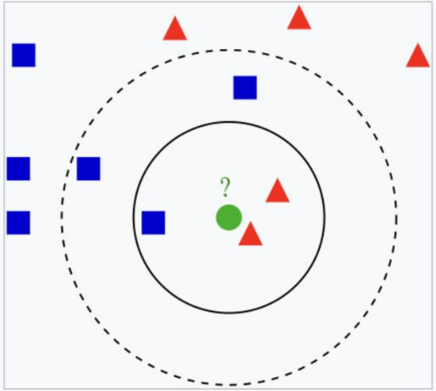
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Our first ML Model

- k-Nearest neighbors (k-NN)
    - Supervised learning
    - Non parametric ***(Distance based method)***
    - Both for Classification and Regression solutions

This is a red triangle  *vs blue square*  *classification problem*



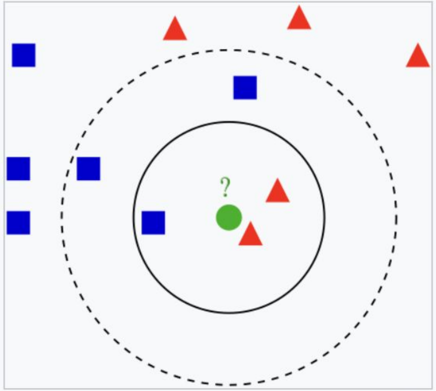
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Our first ML Model

What other classification problem you can think of?

- *Email Spam filter: **y** ∈ { **0,1** }*
- *Face recognition: **y** ∈ { **you, your friend, …** }*
- *Cancer detection: **y** ∈ {**0, 1**}*
- *News topic detection: **y** ∈ { **politics, sports, …** }*
- *Sentiment classification: **y** ∈ {**happy, sad, …**}*



Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

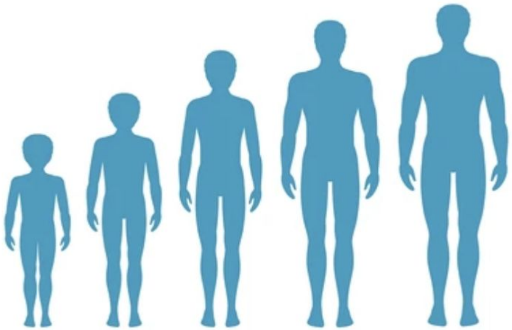
We will be learning today about Regression

- *Insurance cost : $y \in R$*
- *House price: $y \in R$*
- *Weather prediction:* $\boldsymbol{Y \in R}$
- *Energy consumption: $y \in R$*
- *Sales forecasting: $y \in R$*

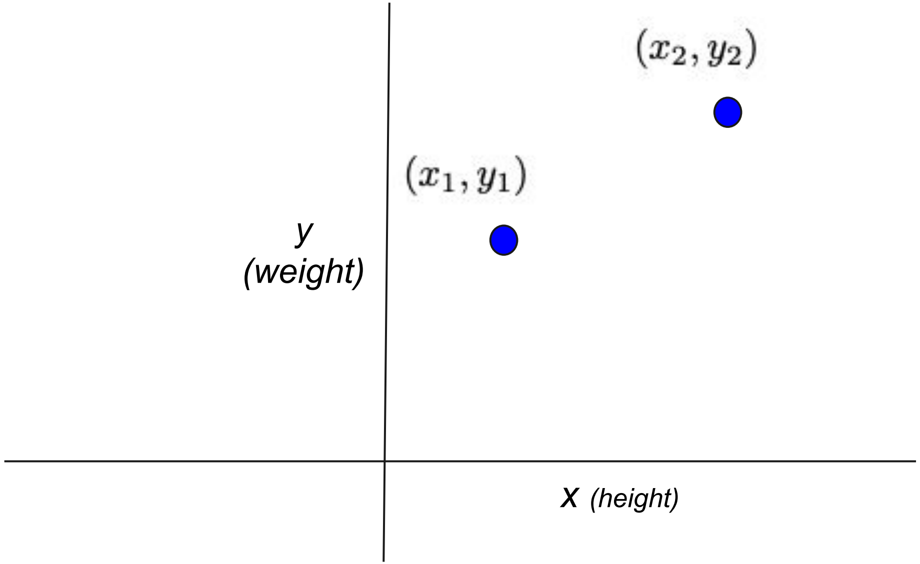
We will be learning today about Regression

- *Person's weight :* $y \in R$

$$f\,(y|x=height)$$

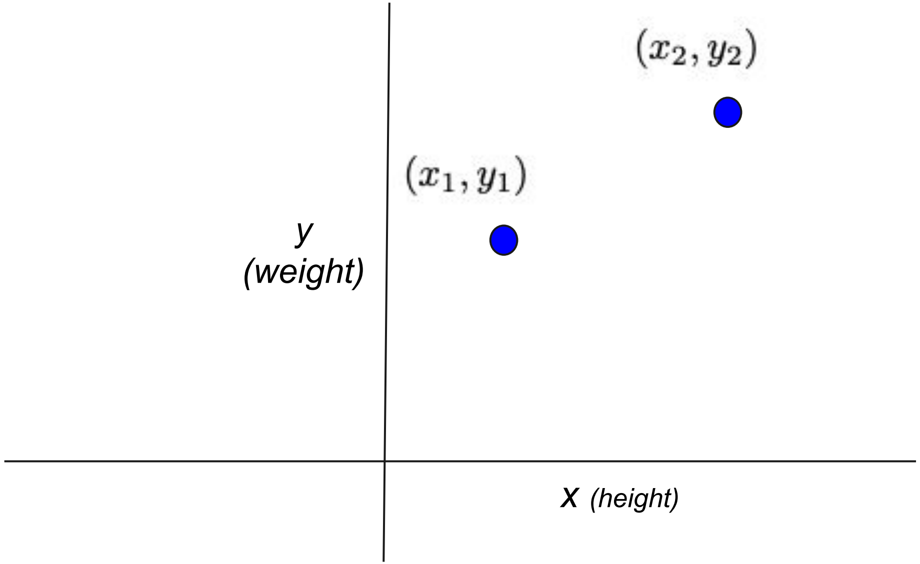


Given these two data points



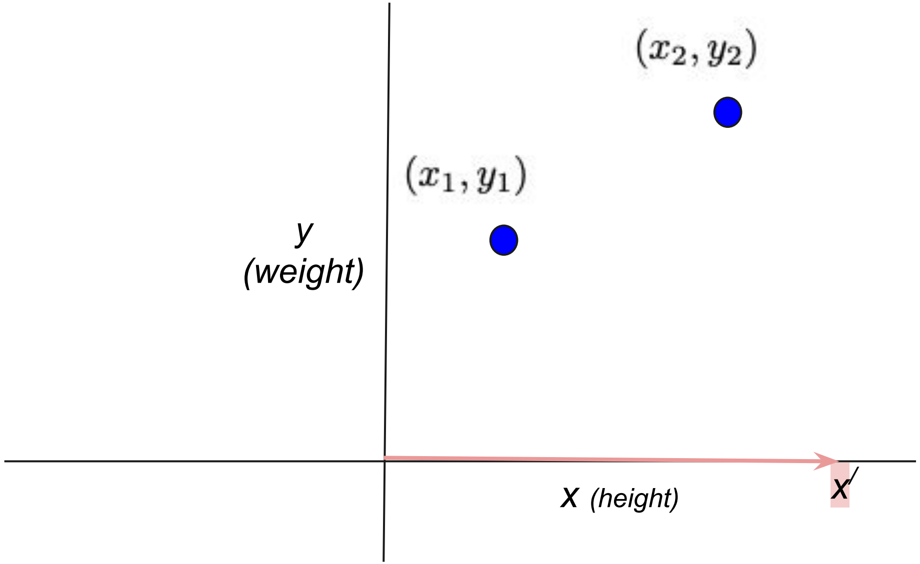
Given two known data points (x_1,y_1), and (x_2,y_2), and

k-NN Regression



Given two known data points (x_1,y_1), and (x_2,y_2), and

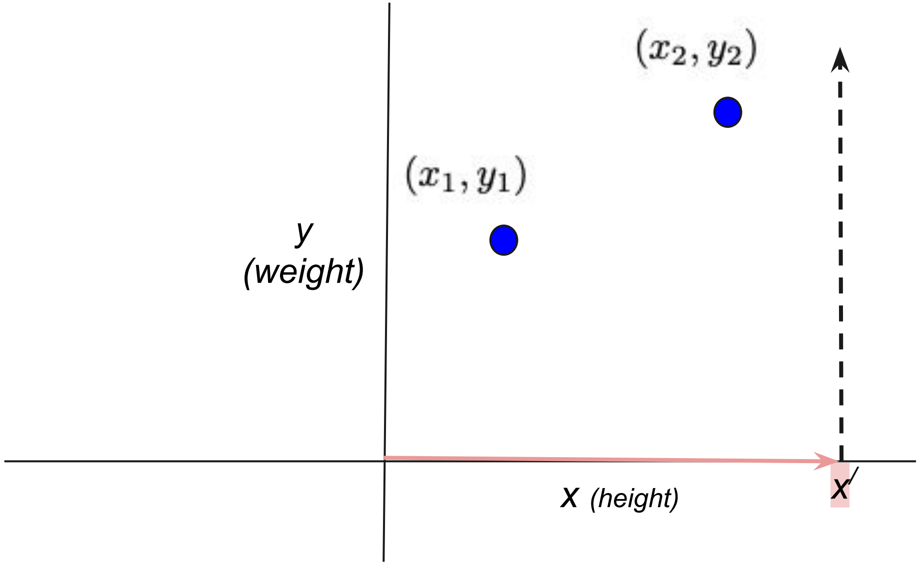
k-NN Regression



Given two known data points (x_1,y_1), and (x_2,y_2), and

- for test input x', you have to predict $y(x')$.
- I.e. you have to plot $(x', ?)$

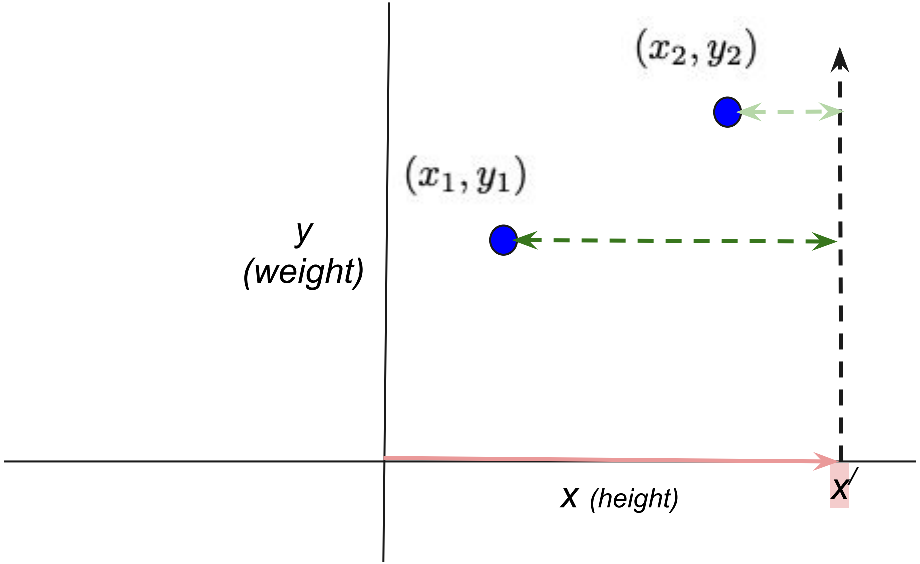
k-NN Regression



Given two known data points (x_1,y_1), and (x_2,y_2), and

- for test input $x^/$, you have to predict $y(x^/)$.
- I.e. you have to plot ($x^/$, ?)
- To estimate the distances let's draw the vertical line

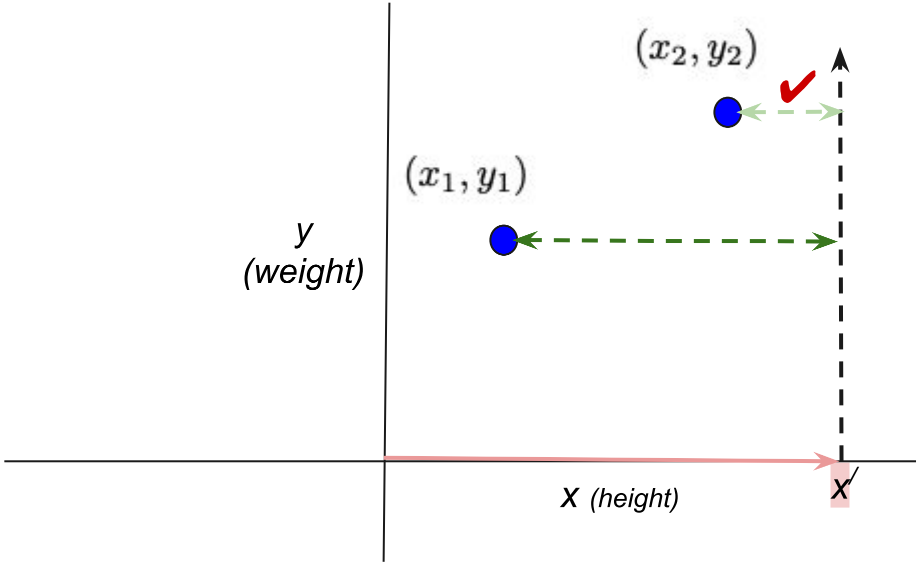
k-NN Regression



Given two known data points (x_1,y_1), and (x_2,y_2), and

- for test input x', you have to predict $y(x')$.
- I.e. you have to plot $(x', ?)$
- To estimate the distances let's draw the vertical line
- Horizontal dotted lines show the point distances (L1)

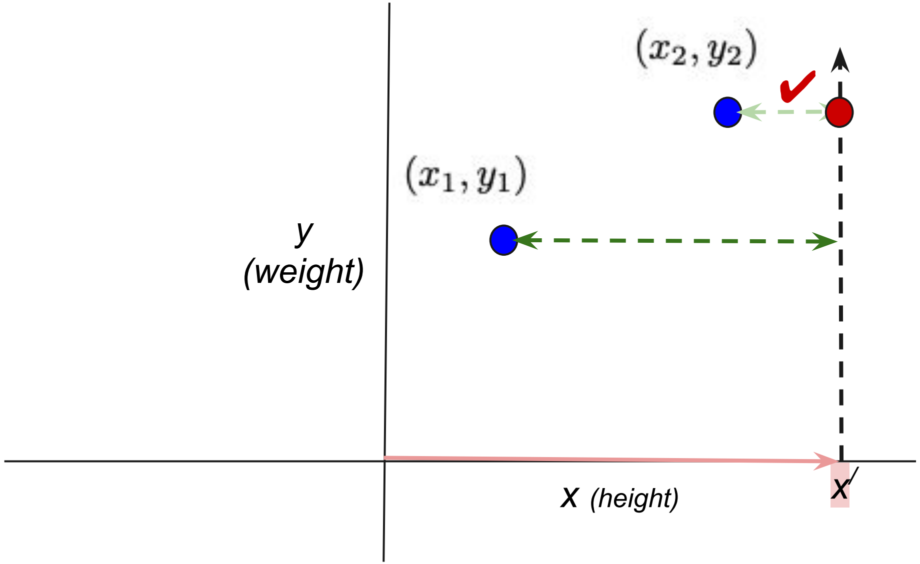
k-NN Regression



Given two known data points (x_1,y_1), and (x_2,y_2), and

- for test input x', you have to predict $y(x')$.
- I.e. you have to plot $(x', ?)$
- To estimate the distances let's draw the vertical line
- Horizontal dotted lines show the point distances (L1)
- We find the lighter green on is the closest one [k(1)-NN]
-

k-NN Regression



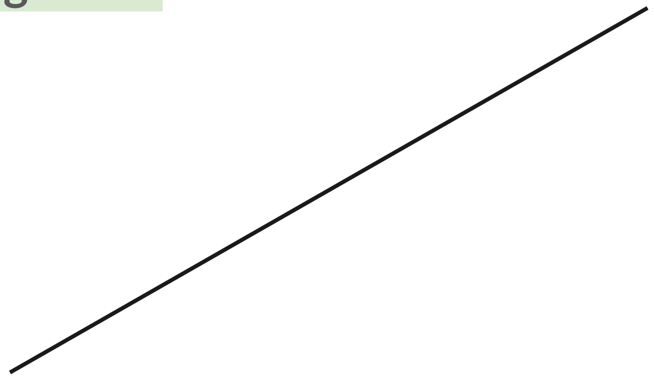
Given two known data points (x_1,y_1), and (x_2,y_2), and

- for test input x', you have to predict $y(x')$.
- I.e. you have to plot $(x', ?)$
- To estimate the distances let's draw the vertical line
- Horizontal dotted lines show the point distances (L1)
- We find the lighter green on is the closest one [k(1)-NN]
- We propagate the associated label(s), i.e.

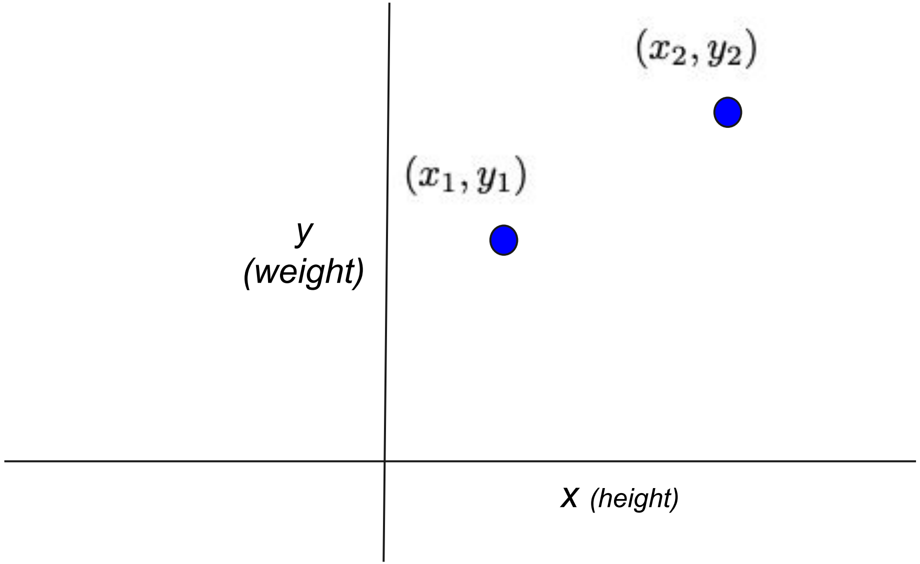
  $y(x') = y_2$
- If we have more data points we may go for a higher k, and take the average

Our Second (Supervised) Model

A simple straight line

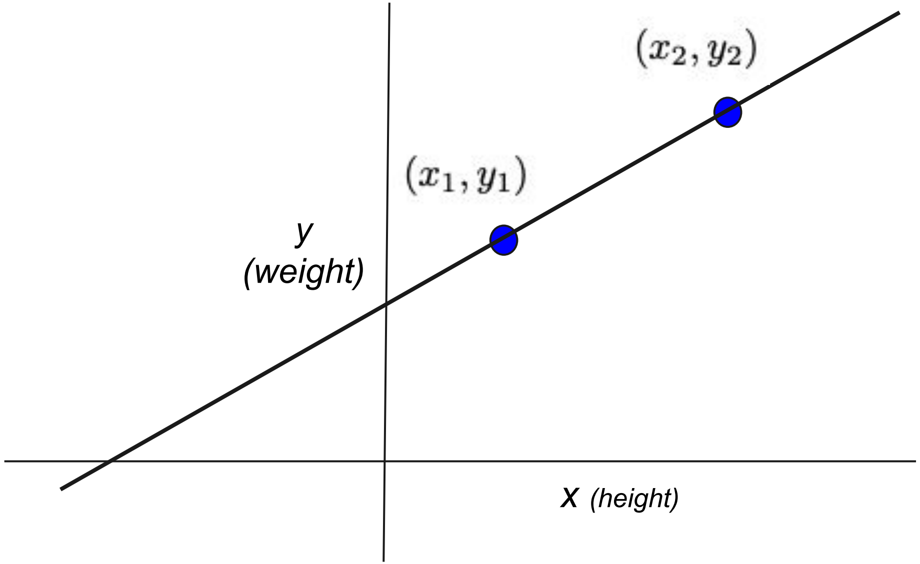


Linear equation, a quick review



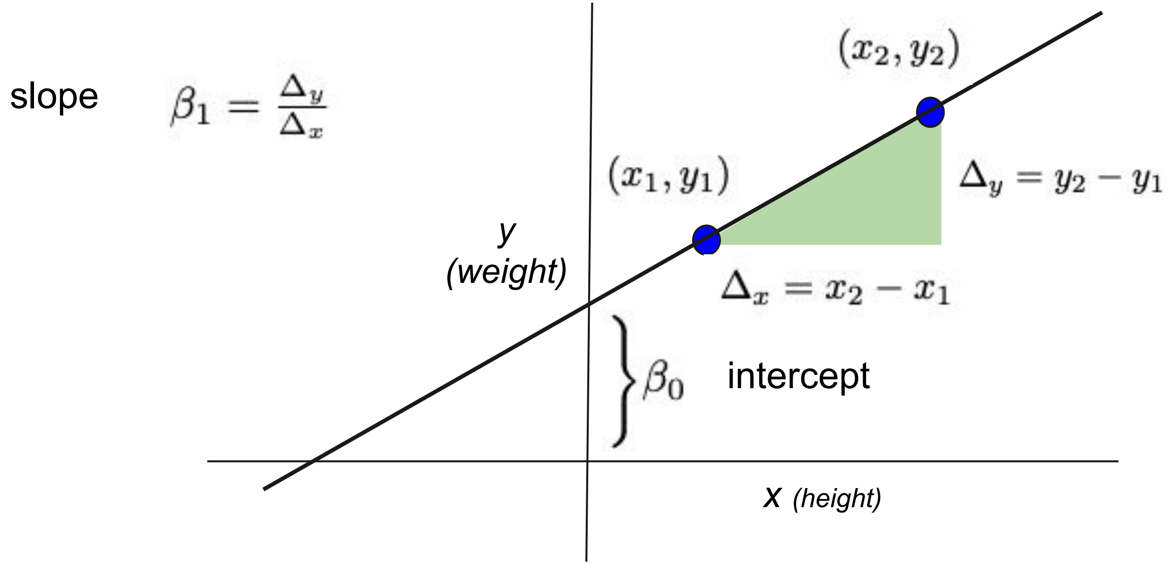
Given two data points

Linear equation, a quick review



We can fit a linear equation

Linear equation to a linear function, a quick review



- This linear equation can be used to explain the relationship between the two axes ***x*** (independent variable) vs ***y*** (dependent variable) - as

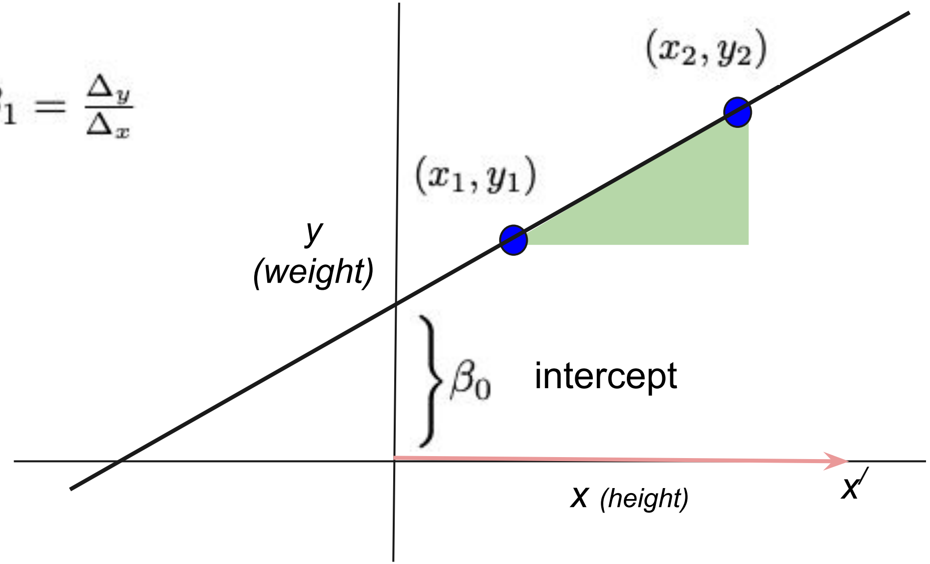
$$y = \beta_0 + \beta_1 x$$

- A simple model with parameters: **slope**, and **intercept**

- For any given x', this model can predict $y(x')$ using the above equation.

Linear equation to a linear function, a quick review

slope $\beta_1 = \frac{\Delta_y}{\Delta_x}$



- This linear equation can be used to explain the relationship between the two axes ***x*** (independent variable) vs ***y*** (dependent variable) - as

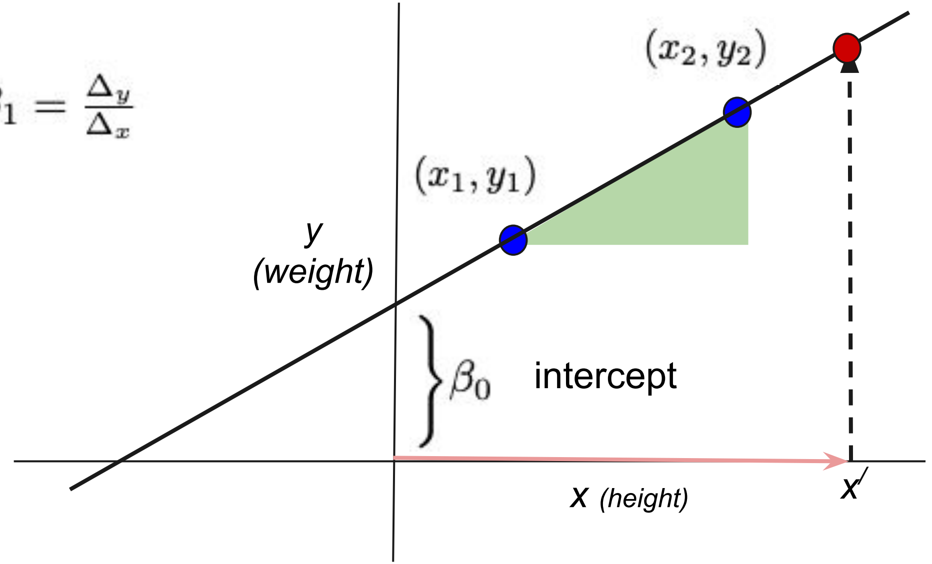
$$y = \beta_0 + \beta_1 x$$

- A simple model with parameters: **slope**, and **intercept**

- For any given x', this model can predict $y(x')$ using the above equation.

Linear equation to a linear function, a quick review

slope $\beta_1 = \frac{\Delta_y}{\Delta_x}$



- This linear equation can be used to explain the relationship between the two axes ***x*** (independent variable) vs ***y*** (dependent variable) - as

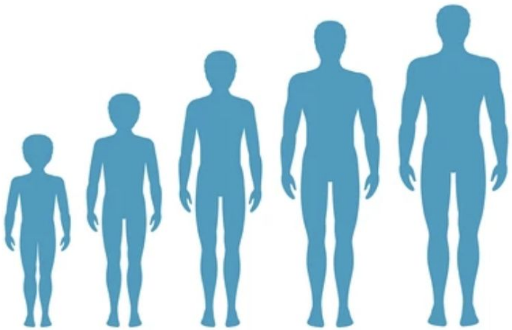
$$y = \beta_0 + \beta_1 x$$

- A simple model with parameters: **slope**, and **intercept**

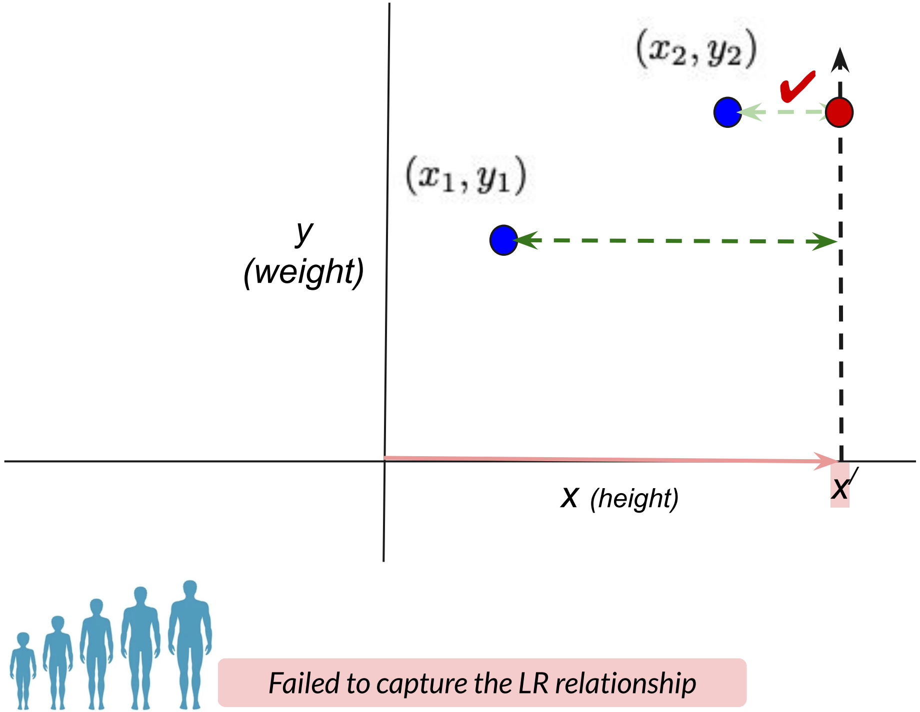
- For any given x', this model can predict $y(x')$ using the above equation.

k-NN vs. Linear Equation

Model (class) comparison!



k-NN Regression



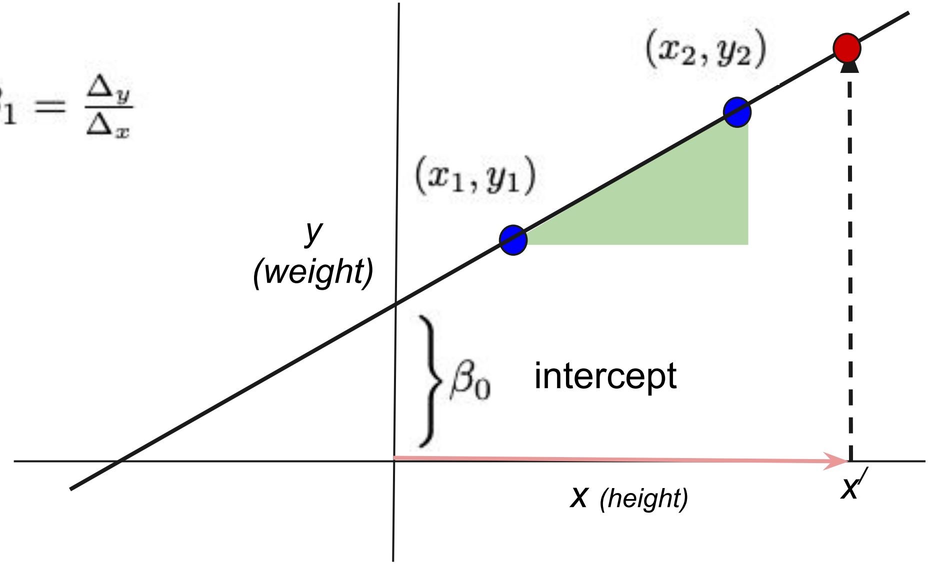
Given two known data points (x_1,y_1), and (x_2,y_2), and

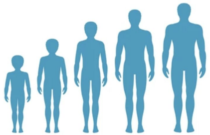
- for test input $x^{/}$, you have to predict $y(x^{/})$.
- I.e. you have to plot $(x^{/}, ?)$
- To estimate the distances let's draw the vertical line
- Horizontal dotted lines show the point distances (L1)
- We find the lighter green on is the closest one [k(1)-NN]
- We propagate the associated label(s), i.e.

  $y(x^{/}) = y_2$
- If we have more data points we may go for a higher k, and take the average

Linear equation to a linear function, a quick review

slope $\beta_1 = \frac{\Delta_y}{\Delta_x}$





Was able to capture the LR relationship

- This linear equation can be used to explain the relationship between the two axes ***x*** (independent variable) vs ***y*** (dependent variable) - as

$$y = \beta_0 + \beta_1 x$$

- A simple model with parameters: **slope**, and **intercept**

- For any given x', this model can predict $y(x')$ using the above equation.

QA