CIS 678 Machine Learning

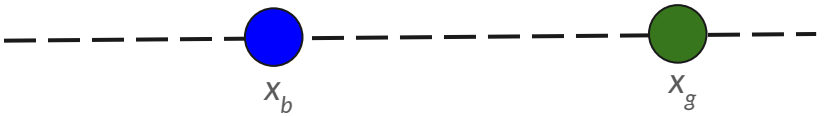
Introduction to Linear Algebra + *k-NN*

Outline

- Proximity vs Distance Metric
- k-NN, our first ML model

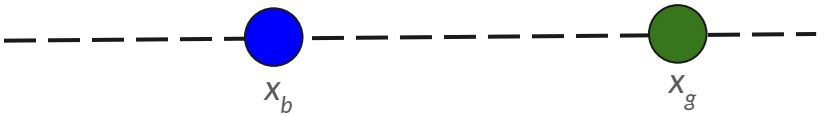
Distance (or Proximity) metric

- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



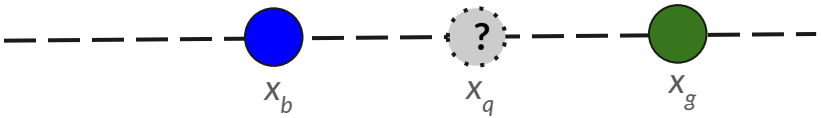
Distance (or Proximity) metric

- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Distance (or Proximity) metric

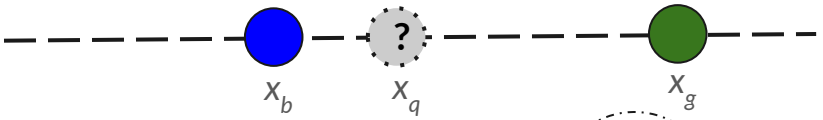
- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Which color do you think ? *is most likely?*

Distance (or Proximity) metric

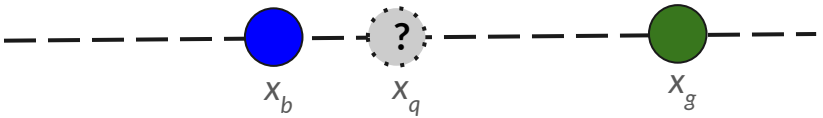
- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Which color do you think ? *is most likely?*

Distance (or Proximity) metric

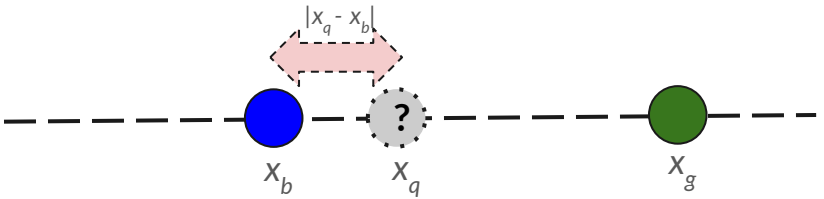
- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Which color do you think ? *is most likely?*

Distance (or Proximity) metric

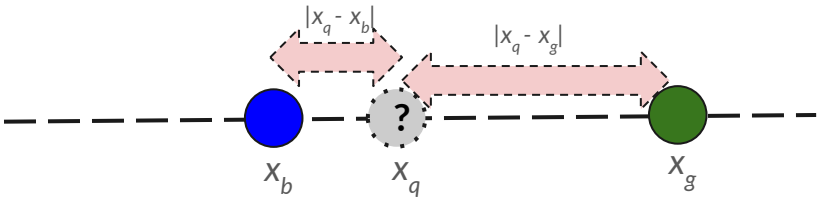
- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Which color do you think ? *is most likely?*

Distance (or Proximity) metric

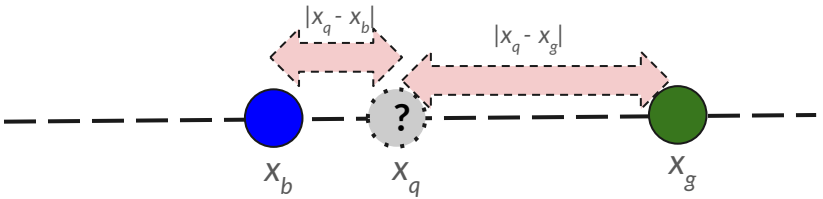
- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



Which color do you think ? *is most likely?*

Distance (or Proximity) metric

- Let's we are give two data points: one, the **blue**, and the other is the **green** circle.



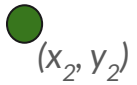
Which color do you think ? *is most likely?*

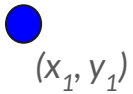
Decision Rule: $|x_q - x_b| < |x_q - x_g|$

Let's Move to Higher Dimensions!

Distance (or Proximity) metric

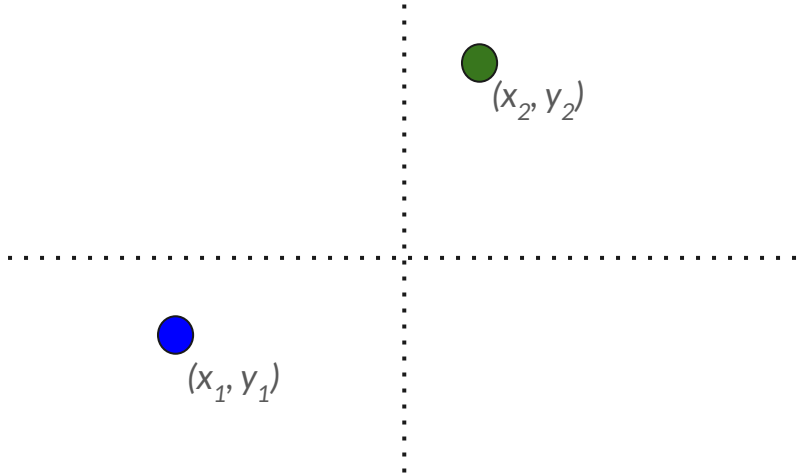
- What is the distance between these two data $p_1(x_1, y_1)$ *and* $p_2(x_2, y_2)$ points (on a 2D plane)?





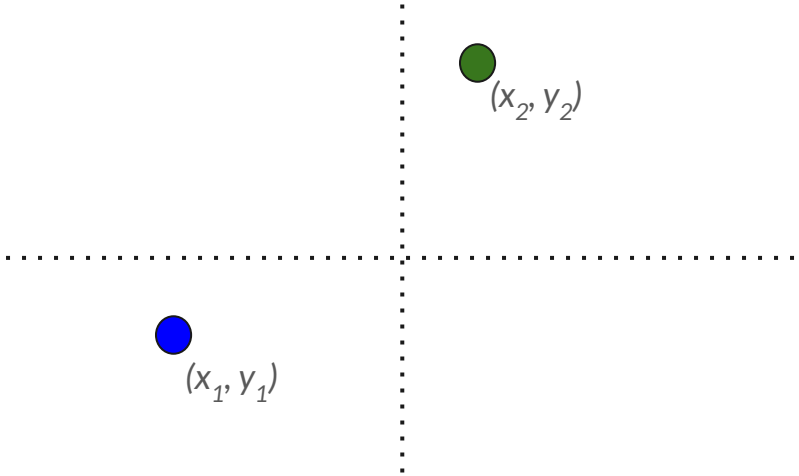
Distance (or Proximity) metric

- What is the distance between these two data $p_1(x_1, y_1)$ *and* $p_2(x_2, y_2)$ points (on a 2D plane)?
- We can use the **Cartesian coordinate system** to quantify the location, and measure their distance.



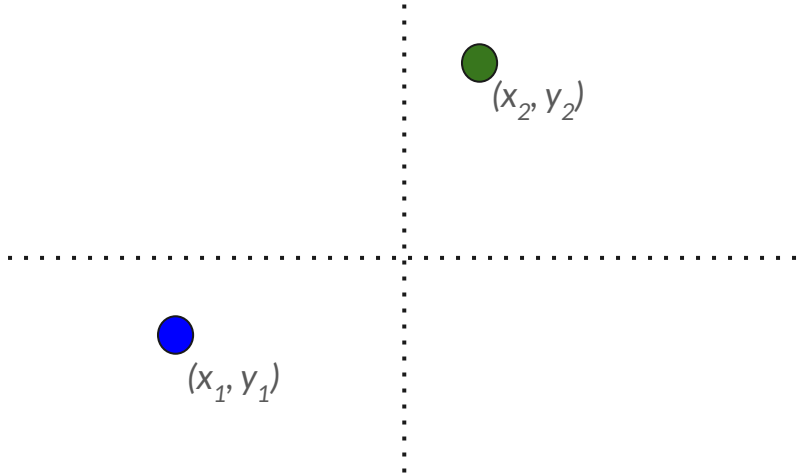
Distance (or Proximity) metric

- What is the distance between these two data $p_1(x_1, y_1)$ *and* $p_2(x_2, y_2)$ points (on a 2D plane)?
- We can use the **Cartesian coordinate system** to quantify the location, and measure their distance.
- We will learn several distance metrics.



Distance (or Proximity) metric

- What is the distance between these two data $p_1(x_1, y_1)$ *and* $p_2(x_2, y_2)$ points (on a 2D plane)?
- We can use the **Cartesian coordinate system** to quantify the location, and measure their distance.
- We will learn several distance metrics.
- Let's start with **L1 distance** metrics also known as **Manhattan distance**

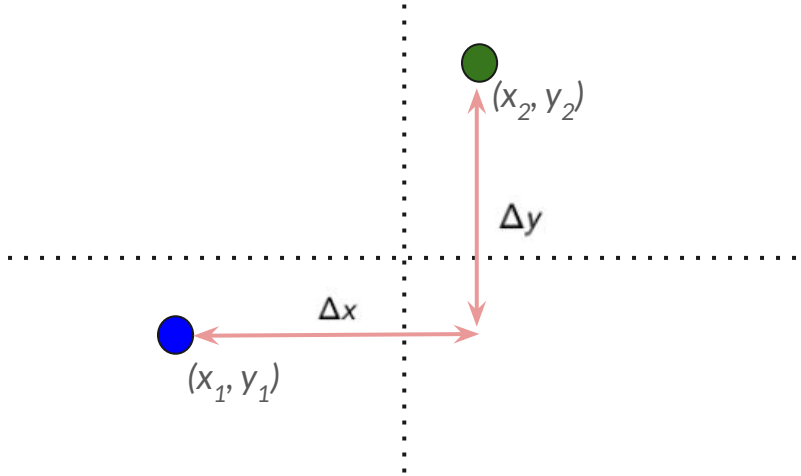


Distance (or Proximity) metric

L1 Distance: The L1 distance between two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ is:

$$\begin{aligned} \text{L1 Distance} &= |x_2 - x_1| + |y_2 - y_1| \\ &= \Delta x + \Delta y \end{aligned}$$

That is, the L1 distance is the sum of the horizontal and vertical sides of the right triangle formed between the two points.

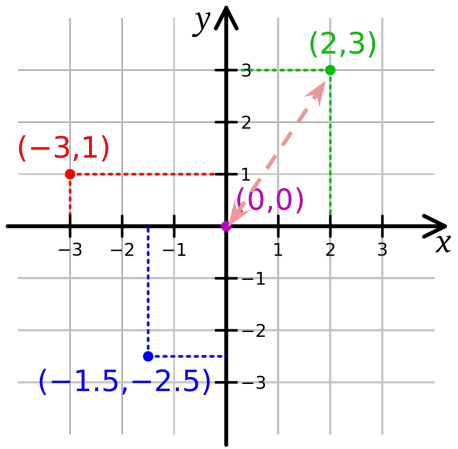


Distance (or Proximity) metric

- **L1 distance**

Let's practice

- L1 distance between vectors [2, 3] and [0, 0] ?



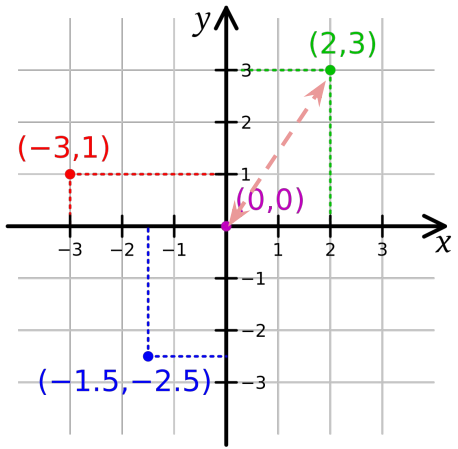
Distance (or Proximity) metric

- **L1** distance, also known as
- **Manhattan** distance

Let's practice

- L1 distance between vectors [2, 3] and [0, 0] ?

  |2-0| + |3-0| = 5

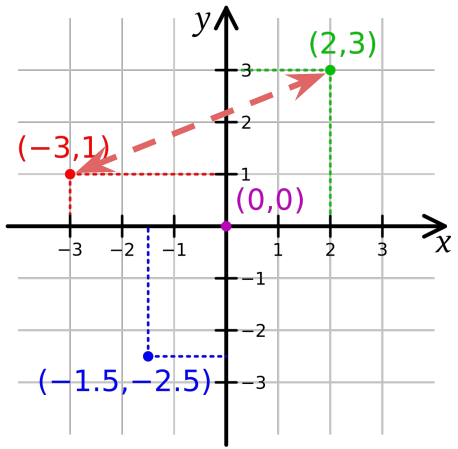


Distance (or Proximity) metric

- **L1** distance, also known as
- **Manhattan** distance

Let's practice

- L1 distance between vectors [2, 3] and [-3, 1] ?



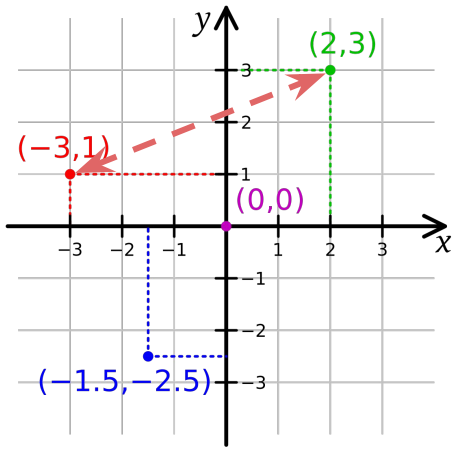
Distance (or Proximity) metric

- **L1** distance, also known as
- **Manhattan** distance

Let's practice

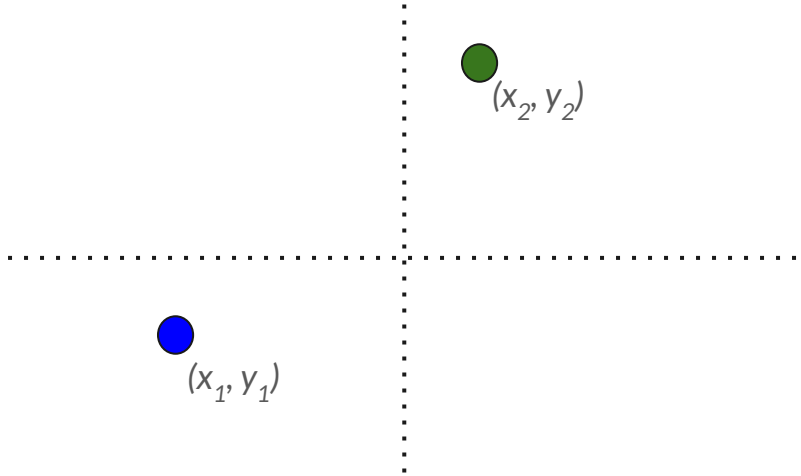
- L1 distance between vectors [2, 3] and [-3, 1] ?

  |2 - (-3)| + |3 - 1| = 5 + 2 = 7



Distance (or Proximity) metric

- What is the distance between these two data points (depicted on a 2D plane)?
- We can use the **Cartesian coordinate system** to quantify the location, and measure their distance.
- We will learn several distance metrics.
- Let's start with **L1 distance** metrics also known as **Manhattan distance**
- Another popular matrics we learned in High School, **L2 distance**, also known as **Euclidean distance**.



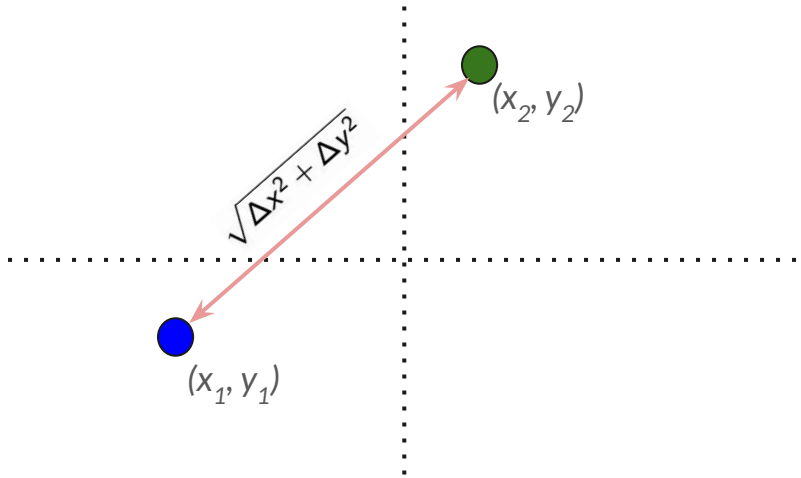
Distance (or Proximity) metric

L2 Distance: The L2 distance between two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ is:

$$\text{L2 Distance} = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$$

$$= \sqrt{\Delta x^2 + \Delta y^2}$$

That is, the L2 distance is the square root of the sum of the squares of the horizontal and vertical sides of the right triangle formed by the two points.



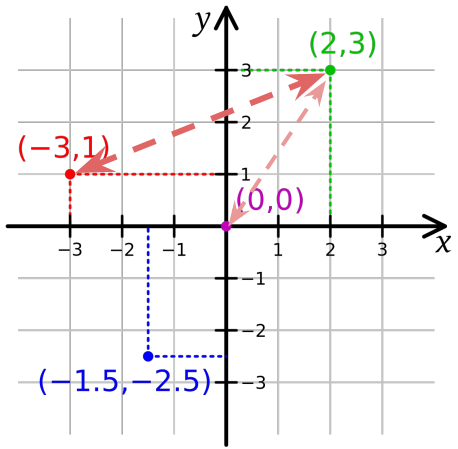
Distance (or Proximity) metric

- **L2 (or Euclidean) distance:**

Let's practice

- L2 distance between vectors [2, 3] and [0, 0] ?

- L2 distance between vectors [2, 3] and [-3, 1] ?



Distance (or Proximity) metric

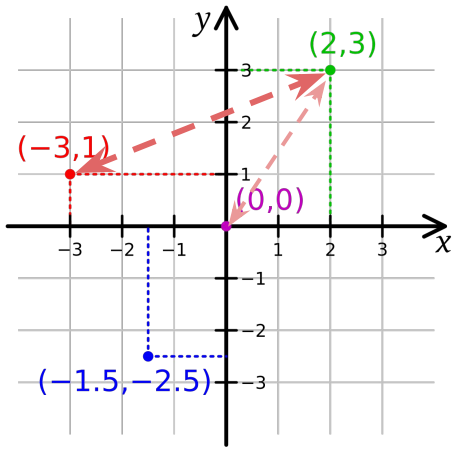
- **L2 (or Euclidean) distance:**

  - L2 distance between vectors [2, 3] and [0, 0] is:

    $$\sqrt{(2-0)^2+(3-0)^2} = \sqrt{13} = 3.61$$

  - L2 distance between vectors [2, 3] and [-3, 1] is:

    $$\sqrt{(2-(-3)^2+(3-1)^2} = \sqrt{29} = 5.39$$

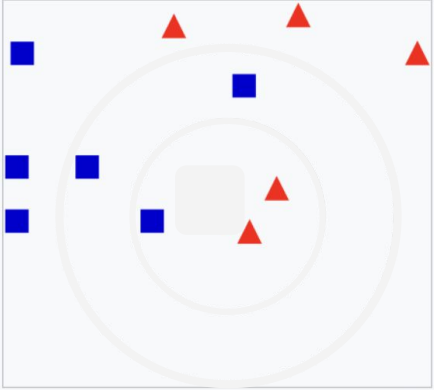


Our first ML Model

- 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

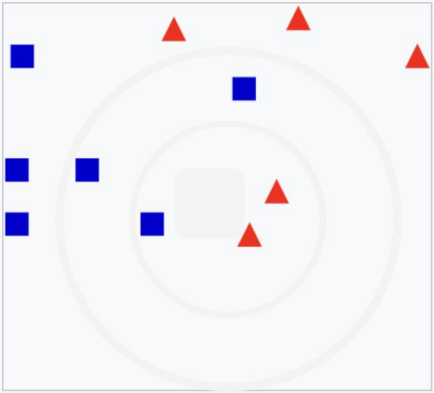


Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it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

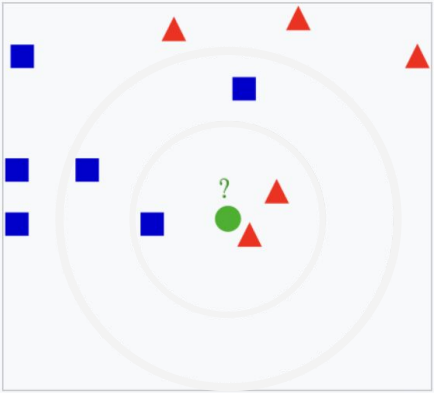


Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it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



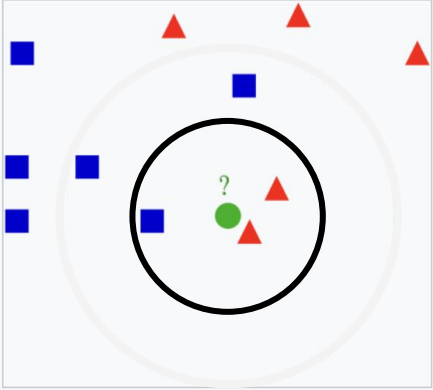
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it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

- For
  - *k = 3, and*
  - *Using L2/Euclidean distance as proximity metric*



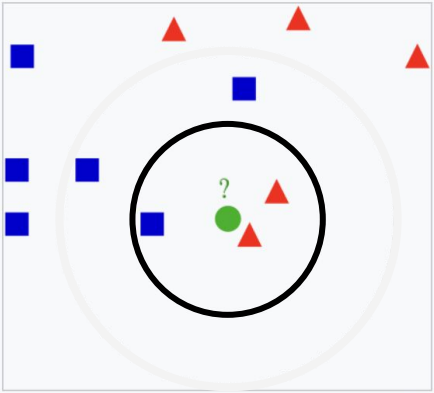
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it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

- For
  - *k = 3, and*
  - *Using L2/Euclidean distance as proximity metric*
  - *The output label would be a Red Triangle:* ▲



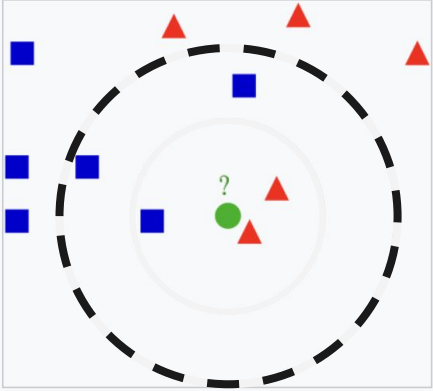
Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If *k = 3* (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* (dashed line circle) it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

- For
  - *k = 5, and*
  - *Using L2/Euclidean distance as proximity metric*
  - *The output label would be a Blue Square:*



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

[1.2.2 Math and Linear Algebra] ....

Another unique Distance metric

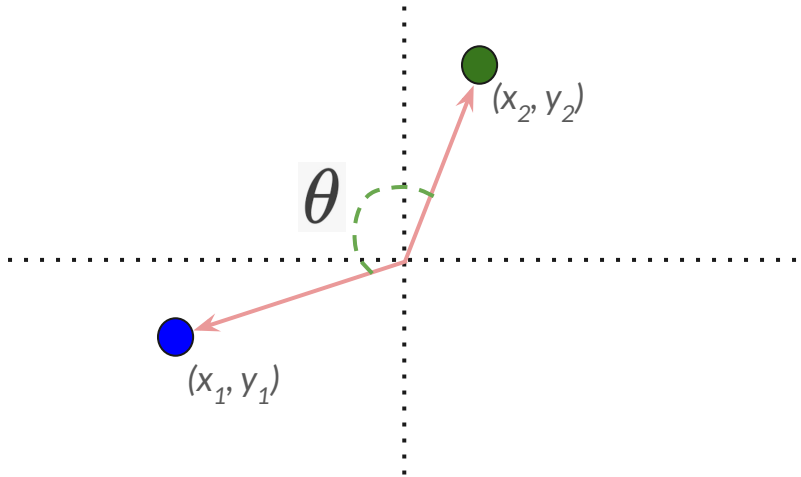
- L1/Manhattan Distance
- L2/Euclidean distance
- **Cosine distance**

Distance (or Proximity) metric

Cosine Distance: The Cosine distance between two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ is:

$$\text{Cosine Distance} = 1 - \frac{p_1^T p_2}{\|p_1\|\|p_2\|}$$

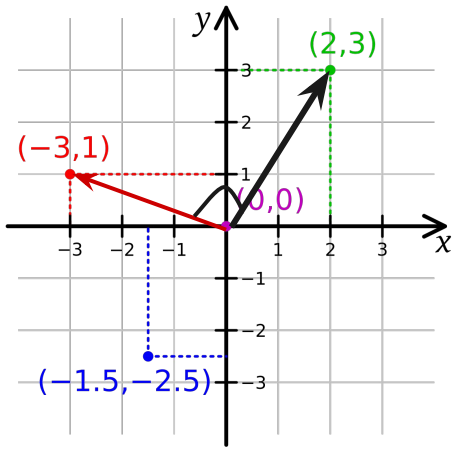
That is, the Cosine distance is the angular distance between two data points.



Cosine distance (angular)

Let's practice

Cosine distance between vectors [2, 3] and [-3, 1] is :

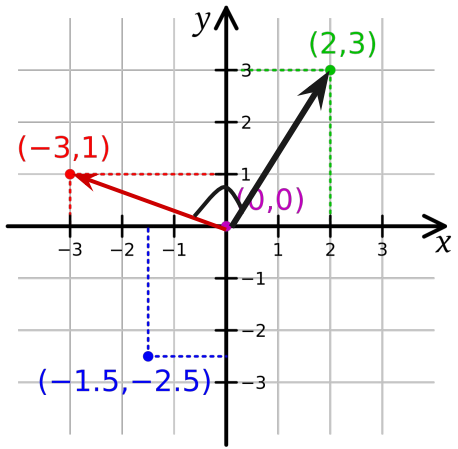


Cosine distance (angular)

Let's practice

Cosine distance between vectors [2, 3] and [-3, 1] is :

= 1 - (-0.26)
= 1.26



Comparing Distances

Distance Ranges:

- **L1/Manhattan Distance**: [0 - ∞]
- **L2/Euclidean Distance**: [0 - ∞]
- **Cosine Distance**: [0 - 2]

We will explore their advantages and disadvantages as the course progresses.

QA