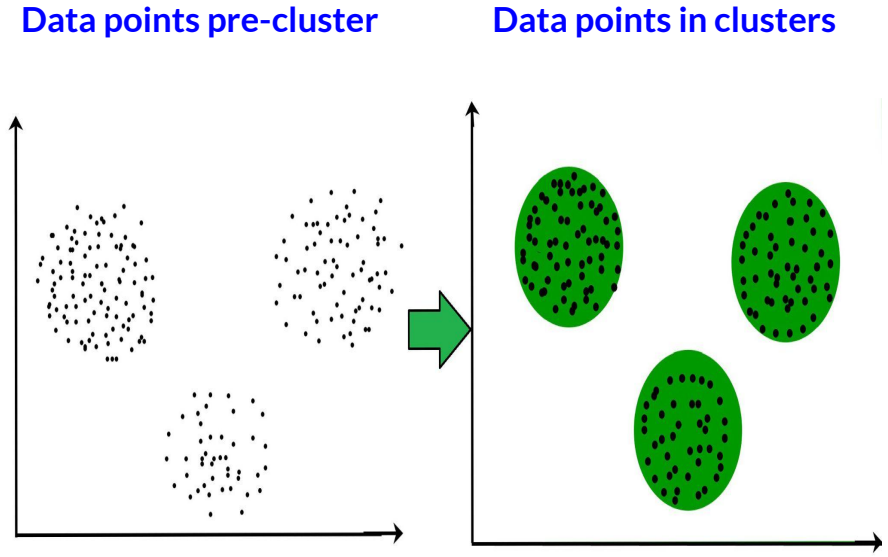
Measuring Distance Between Data Points

- Why measure distance?
- Manhattan
- Euclidean
- Hamming Distances
- Selecting a model

Why measure distance?

- In data science and machine learning, we compare data points in different manners. Data points are rarely equal but similar enough to be put into a cluster or classification. Clusters are needed for unsupervised learning, where we do not know the correct answers in advance.
- In contrast, classification is used in supervised learning, where each data point has a known label or category. The goal is to train a model to recognize patterns in the labeled data so it can predict the label of new, unseen data points.
- For example, if we have emails labeled as "spam" or "not spam" a classification model can learn from these examples and start identifying new spam emails automatically.
- Distance measurements play a key role in classification algorithms like K-Nearest Neighbors (KNN), which assign a class to a data point based on how close it is to other labeled points.

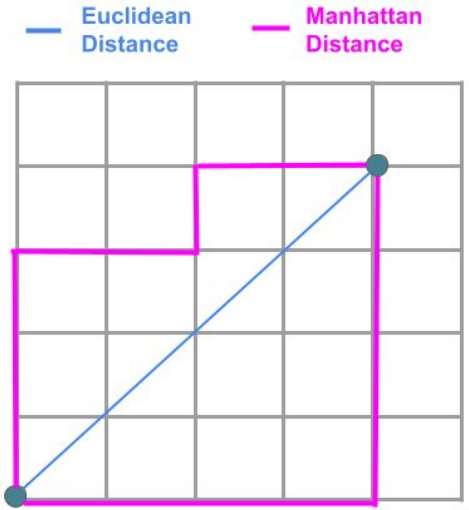


In short: Any machine learning model needs data. By measuring the data we are able to separate said data into meaningful data for the different models out there.

Manhattan & Euclidean distance

Manhattan distance is a metric used to determine the distance between two points in a grid-like path. Unlike Euclidean distance, which measures the shortest possible line between two points, Manhattan distance measures the sum of the absolute differences between the coordinates of the points. This method is called "Manhattan distance" because, like a taxi driving through the grid-like streets of Manhattan, it must travel along the grid lines. *

In this example with coordinates (0,0) to (4,4) the Euclidean distance is 4, whereas Manhattan is 8.



* https://www.datacamp.com/tutorial/manhattan-distance

Hamming distance

- Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, hamming distance is the number of bit positions in which the two bits are different.
- The most famous use case for hamming distance is error correction. We all know that computers are just 1s and 0s. When sending data the message may be altered or corrupted in a way not intended. With hamming distance we are able to correct for these problems, provided we design our codes so that each valid code is at least 2-3 bits different.
- This is great for fixing 1 bit errors but fails in more complex errors. Other methods are used when dealing with such problems but these are out of scope for this presentation.


```
0000
1111
1010
0101
```

Given these 4 valid codes. Suppose we receive a 1110 code which is invalid. We can calculate the hamming distance between the received code and any valid code.

Codeword	Distance from 1110
0000	3
1111	1 ✅
1010	2
0101	3

Since 1111 is only 1 bit away from the original message we can correct the error with a very high success rate.

Selecting a model

Measuring data and putting it into categories, clusters or whatever we do with it needs a purpose. The purpose is to prepare the data for a certain model that fits our need.
Classification/Regression are supervised learning. Clustering and dimensionality reduction are unsupervised learning.