11 K-Means Clustering

Created Date: 2022-11-15

> Metadata 🗃️
>
> - Title: K-Means Clustering (Unsupervised Learning)
> - Author: Andrew Jones
> - Reference: Data Science Infinity

> Links & Tags 🔗
>
> - Index: Course Note Index
> - Atomic Tag: #datascience
> - Subatomic Tags: #machinelearning #k-means #unsupervisedlearning

---

High-Level Overview

Jupyter Notebook: Basic K-Means Clustering Template

> K-Means Clustering partitions data points into distinct groups based on their similarity with each other. The number of distinct groups is determined by the value *k*.

- Common in customer segmentation analysis
- Once a value for *k* is established, the algorithm completes the following steps;
  - Selects *k* random points in space (called centroids)
  - Measures distance between each data point and each centroid, assigning each data point to the nearest centroid

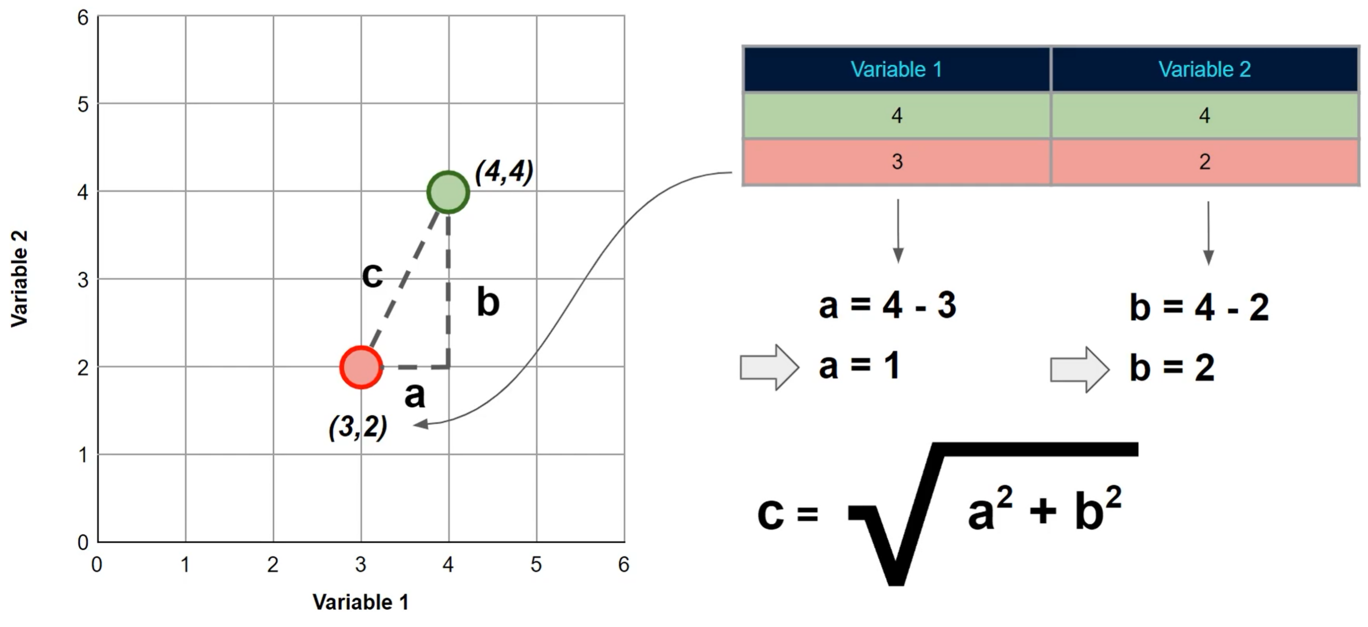
- Repositions the centroids to the mean dimension values of it's cluster
- Reassigns each data point to the nearest centroid
- *The algorithm will repeat the above two steps (3 and 4) until no data points move between clusters*

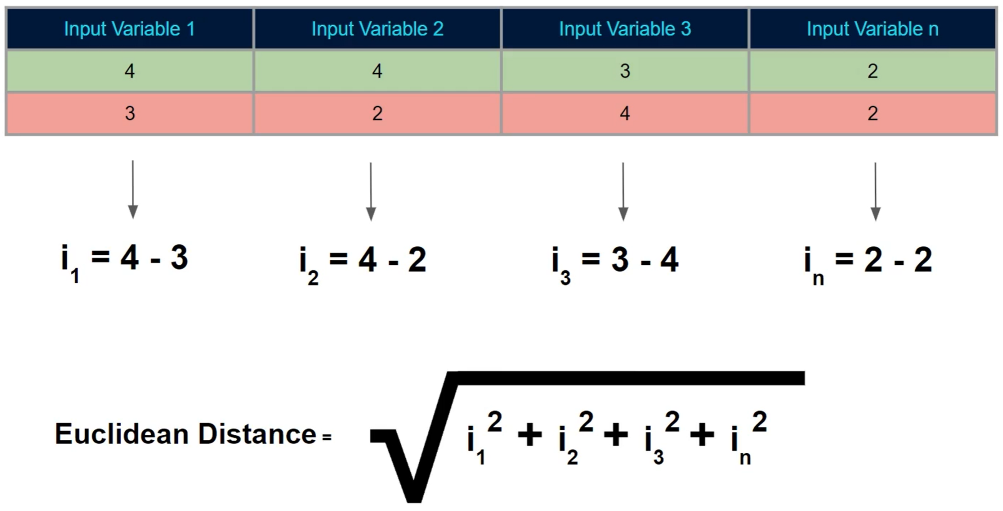
Advanced Theory

Jupyter Notebook: Advanced K-Means Clustering Template

Measuring Distances in Multi-Dimensional Space

- Distance is determined by the Euclidean distance between input variables
- This distance is calculated using Pythagorean theorem as the distance between any two points is the hypotenuse (c)





Input Variable 1	Input Variable 2	Input Variable 3	Input Variable n
4	4	3	2
3	2	4	2

$$i_1 = 4 - 3 \quad i_2 = 4 - 2 \quad i_3 = 3 - 4 \quad i_n = 2 - 2$$

$$\text{Euclidean Distance} = \sqrt{i_1^2 + i_2^2 + i_3^2 + i_n^2}$$

Formula

- Euclidean Distance: $c = \sqrt{i_1^2 + i_2^2 + i_n^2}$
  - Where;
  - $i_n = q1 - p1$
  - Since we're taking the square root of these values, it doesn't matter the order as the result will always be positive

Importance of Feature Scaling

> Feature Scaling is where we force the values from different columns to exist on the same scale, in order to enhance the learning capabilities of the model. The two most common techniques are *Standardization* and *Normalization*.

- It's important to consider feature scaling when measuring distance between variables
- Standardization rescales data to have a mean of 0 and a standard deviation of 1
- Normalization rescales data so that it exists in a range between 0 and 1
- Normalization is more appropriate for the KNN algorithm
  - Comparable to categorical variables
  - Implements the same range for all variables

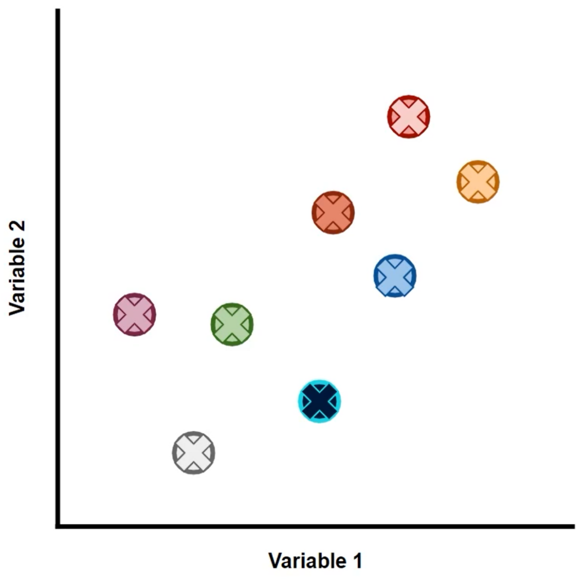
Formulas

- $X_{Standardized} = \frac{(X - mean(X))}{Std.Deviation(X)}$
- $X_{Normalized} = \frac{(X - min(X))}{max(X) - min(X)}$

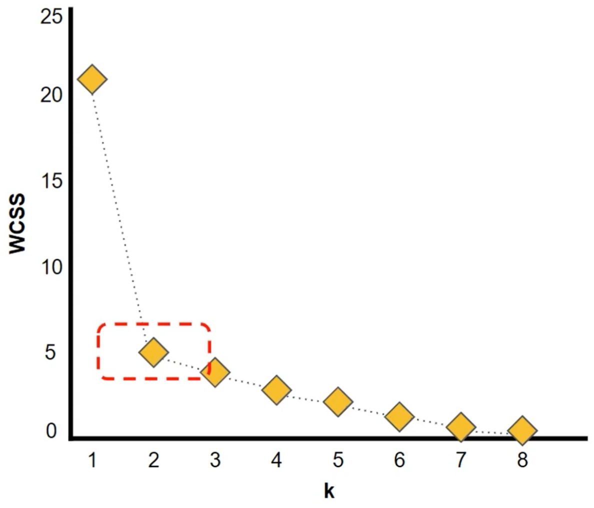
What Value for K?

> Within Cluster Sum of Squares (WCSS) measures the sum of squared Euclidean distances that data points lie from their closest centroid. It can help us understand the point where adding more centroids provides little benefit.

- Choosing a *k* value for clustering differs from KNN
- WCSS will decrease as we increase the value of *k*
- The optimal *k* value is the value with a prominent decrease in WCSS, and before diminishing returns
- We sum the squares of Euclidean distances because the original formula for Euclidean distances calculates the square root
- We can start with *k* = *1* and calculate the distance between each data point and the one centroid, plotting it for visualization
- We then add a centroid (*k* = *2*) and re-calculate the distances between the clustered data points to its centroid, again plotting for visualization
- This process is repeated until we find the optimal value for *k*
- WCSS is referred to as Inertia in Python



WCSS measures the sum of squared euclidean distances that data points lie from their closest centroid


WCSS
k