**UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**



**Instructor: Dr. Lê Văn Vinh, Ph. D**

**Project 3**

**PROJECT REPORT**

**Agglomerative Clustering**

**Trần Minh Quang - 17110067**

**SCHOOL YEAR: 2020 – 2021**

**HO CHI MINH CITY – January 2021**

Table of Contents

[1. Introduction 2](#_Toc61907198)

[2. Agglomerative Clustering 1](#_Toc61907199)

[2.1. How Agglomerative Clustering works 1](#_Toc61907200)

[2.2. Dendrograms 3](#_Toc61907201)

[2.3. Linkage criteria 4](#_Toc61907202)

[2.4. Distance Metric 5](#_Toc61907203)

[3. Data processing 1](#_Toc61907204)

[3.1. Dataset 1](#_Toc61907205)

[3.2. Process data 2](#_Toc61907206)

[4. Execution 1](#_Toc61907207)

[5. Evaluation 1](#_Toc61907208)

[References 2](#_Toc61907209)

# Introduction

In machine learning, unsupervised learning is a machine learning model that infers the data pattern without any guidance or label. Many models are included in the unsupervised learning family, but the one we are discussing here is Agglomerative Clustering.

Agglomerative Clustering or bottom-up clustering essentially started from an individual cluster (each data point is considered as an individual cluster, also called leaf), then every cluster calculates their distance with each other. The two clusters with the shortest distance with each other would merge creating what we called node. Newly formed clusters once again calculating the member of their cluster distance with another cluster outside of their cluster. The process is repeated until all the data points assigned to one cluster called root. The result is a tree-based representation of the objects called dendrogram.

# Agglomerative Clustering

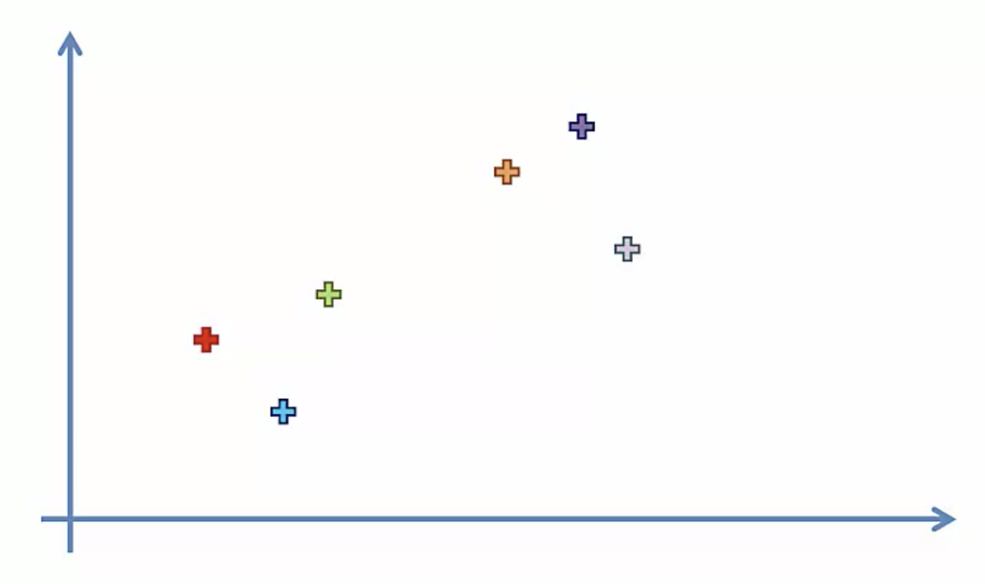
## How Agglomerative Clustering works

Hierarchical clustering algorithms group similar objects into groups called clusters. There are two types of hierarchical clustering algorithms:

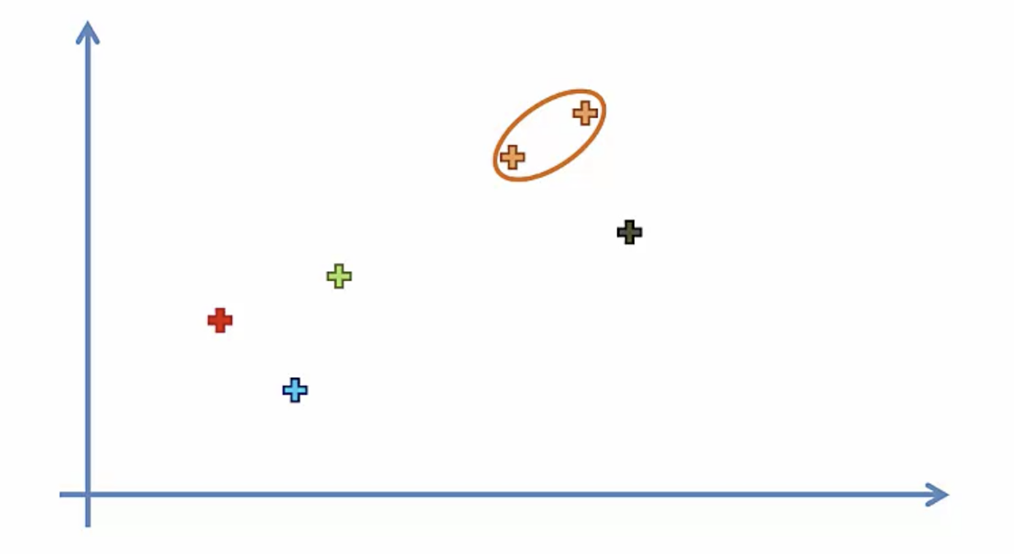
Agglomerative uses Bottom-up approach. It starts with many small clusters and merge them together to create bigger clusters.

#### Steps to perform Agglomerative Clustering:

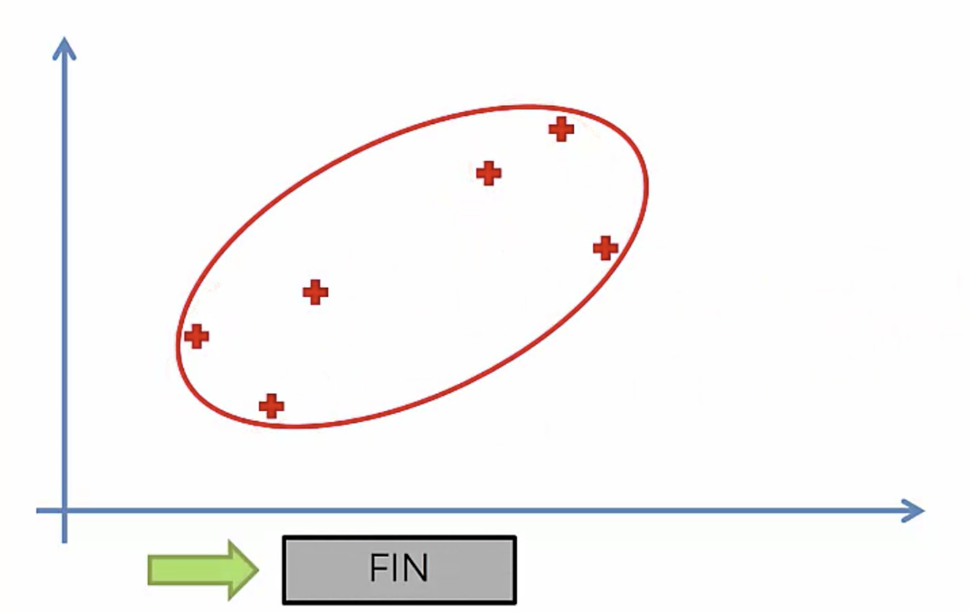
1. Make each data point a cluster



1. Take the two closest clusters and make them one cluster



1. Repeat step 2 until there is only one cluster

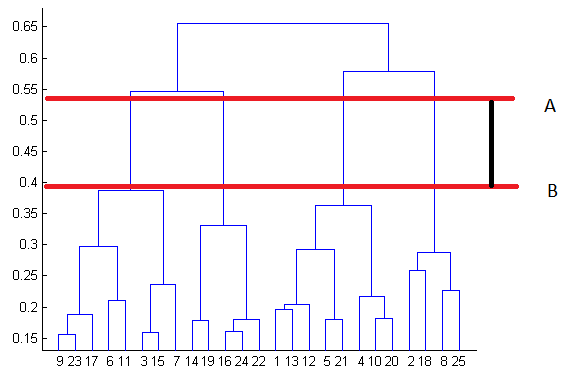


## Dendrograms

We can use a dendrogram to visualize the history of groupings and figure out the optimal number of clusters.

* Determine the largest vertical distance that doesn’t intersect any of the other clusters
* Draw a horizontal line at both extremities
* The optimal number of clusters is equal to the number of vertical lines going through the horizontal line

For example, in the case below, the optimal number of clusters is 4.



## Linkage criteria

Similar to gradient descent, you can tweak certain parameters to get drastically different results. The linkage criteria refers to how the distance between clusters is calculated.

#### Single Linkage

The distance between two clusters is the shortest distance between two points in each cluster.

#### Complete Linkage

The distance between two clusters is the longest distance between two points in each cluster.

#### Average Linkage

The distance between clusters is the average distance between each point in one cluster to every point in other cluster.

#### Ward Linkage

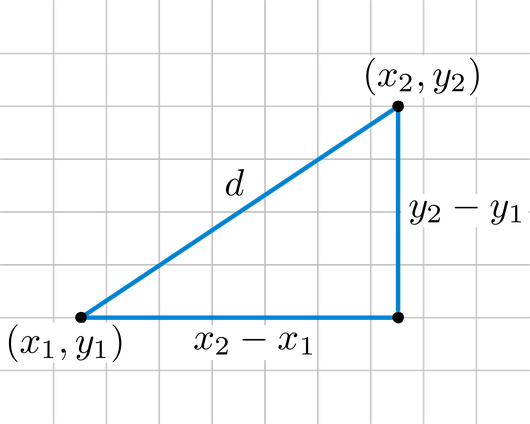
The distance between clusters is the sum of squared differences within all clusters.

## Distance Metric

There are a few ways to calculate the distance between data points, but for this project we’ll only use the Euclidean Distance.

#### Euclidean Distance

The shortest distance between two points. For example, if x = (a, b) and y = (c, d), the Euclidean distance between x and y is √ ((a - c) ² + (b - d) ²)



# Data processing

## Dataset

This data set includes 721 Pokemon, including their number, name, first and second type, and basic stats: HP, Attack, Defense, Special Attack, Special Defense, and Speed.

The data:

#: ID for each Pokemon

Name: Name of each Pokemon

Type 1: Each Pokemon has a type, this determines weakness/resistance to attacks

Type 2: Some Pokemon are dual type and have 2

Total: Sum of all stats that come after this, a general guide to how strong a Pokemon is

HP: Hit points, or health, defines how much damage a Pokemon can withstand before fainting

Attack: The base modifier for normal attacks

Defense: The base damage resistance against normal attacks

SP Atk: Special attack, the base modifier for special attacks

SP Def: The base damage resistance against special attacks

Speed: Determines which Pokemon attacks first each round

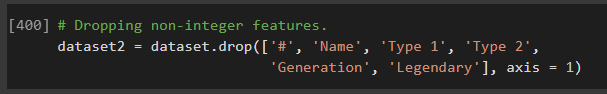
Generation: The game version that a Pokemon is introduced

Legendary: Some Pokemon has a stronger Legendary version of themselves

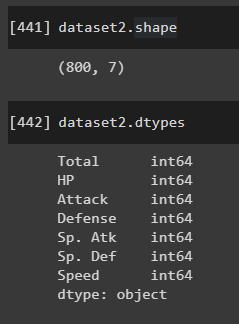
## Process data

#### Filter out non-interger features

Attributes that are non-interger will be dropped to focus on the Pokemon’s stats.



The dataset after filtering:



#### Reduce the dimensionality of the dataset using t-Distributed Stochastic Neighbour Embedding(t-SNE)

##### Goals for reducing the dimensionality of the data

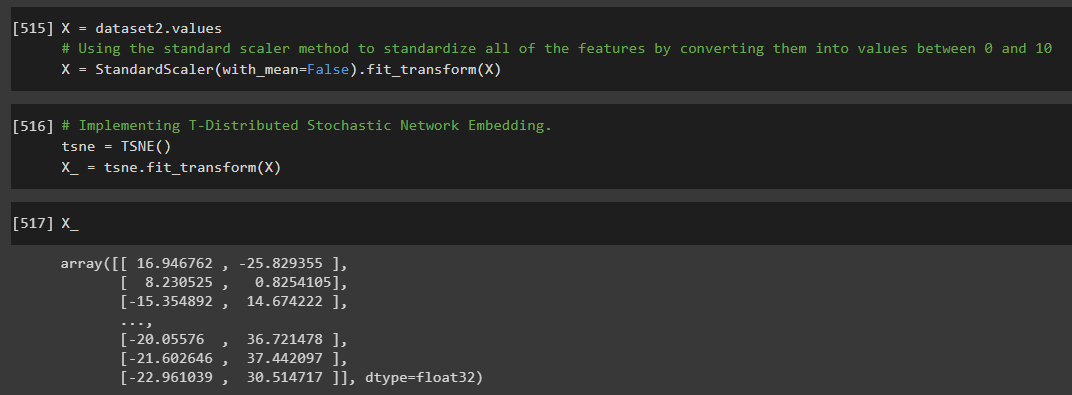
* Preserve as much significant structure or information of the data present in the high-dimensional data as possible in the low-dimensional representation.
* Increase the interpretability of the data in the lower dimension
* Minimizing information loss of data due to dimensionality reduction

##### t-Distributed Stochastic Neighbourh Embedding(t-SNE)

* An unsupervised, randomized algorithm, used only for visualization
* Applies a non-linear dimensionality reduction technique where the focus is on keeping the very similar data points close together in lower-dimensional space.
* Preserves the local structure of the data using student t-distribution to compute the similarity between two points in lower-dimensional space.
* t-SNE uses a heavy-tailed Student-t distribution to compute the similarity between two points in the low-dimensional space rather than a Gaussian distribution, which helps to address the crowding and optimization problems.

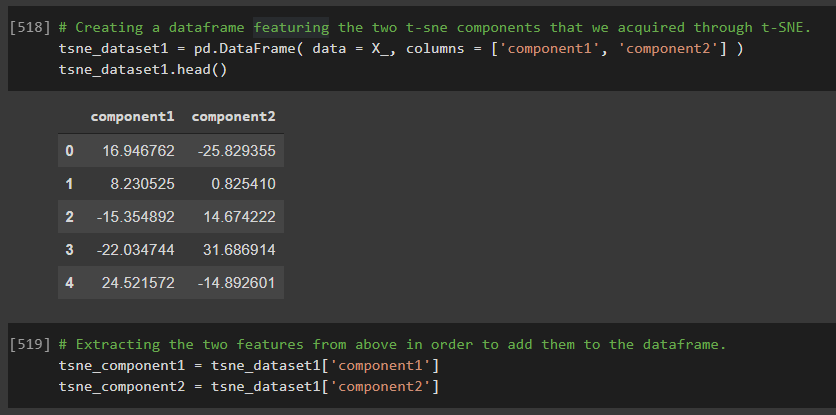
##### Perform dimensionality reduction

We transform the dataset into a 2-dimensional array for easier visualization.

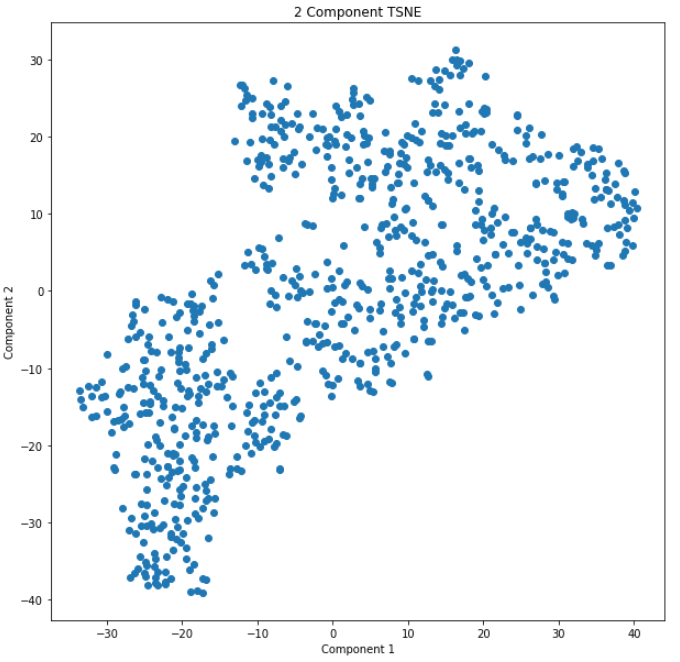


# Execution

#### Creating a dataframe

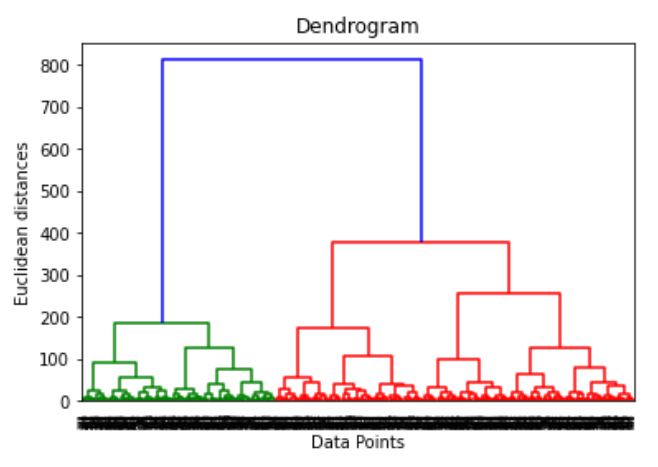


#### Visualizing the DataFrame



#### Dendrogram

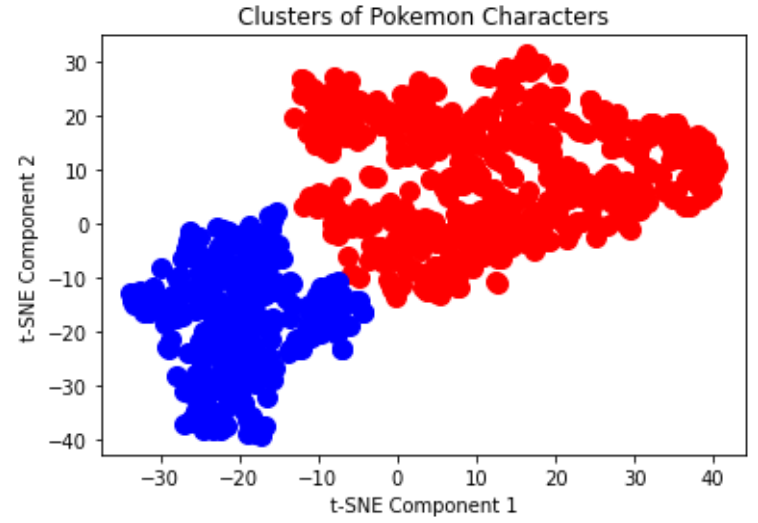
We use Euclidean distance to measure the distance between each data points



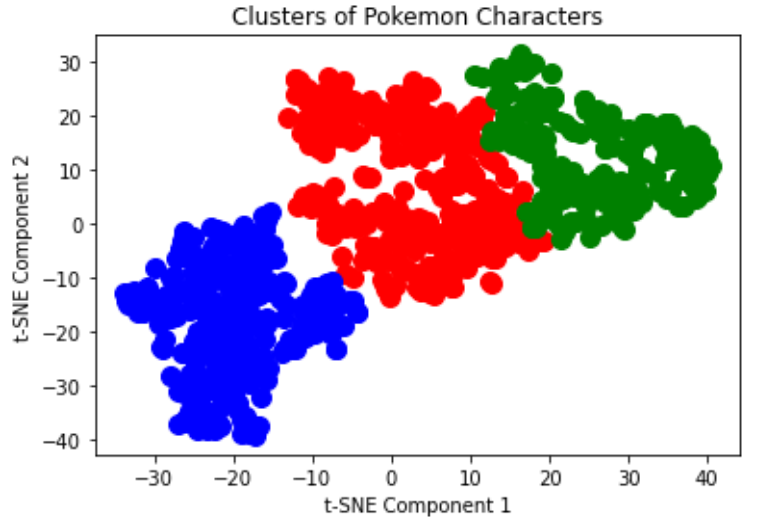
Because the data points are uniformly distributed, it is hard to cluster the data set. But using the Dendrogram, we can try spliting the dataset into different amount of clusters.

We will do multiple tries, so I only work with Ward linkage to calculate the distance between clusters.

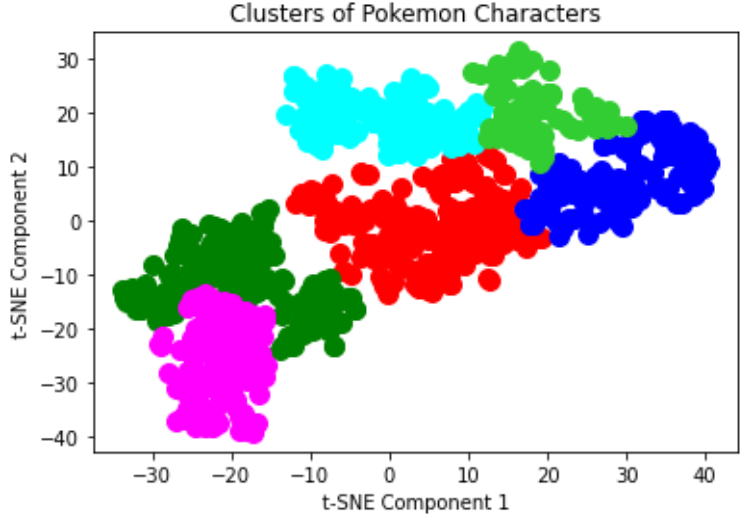
##### 2 clusters



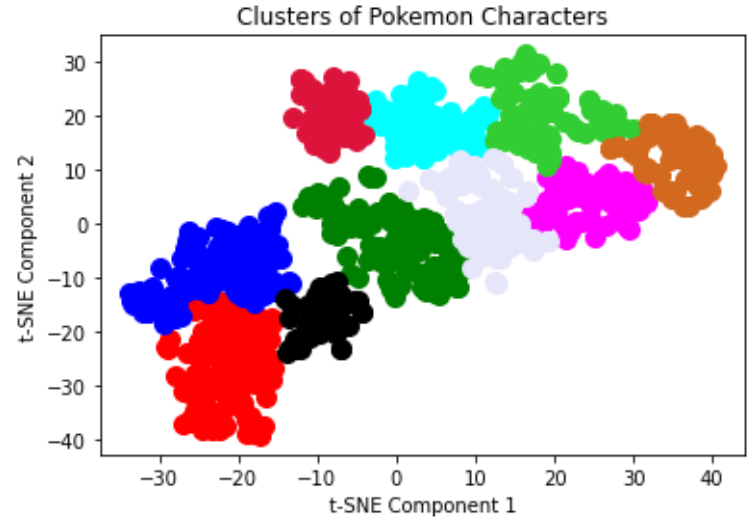
##### 3 clusters



##### 6 clusters



##### 10 clusters



# Evaluation

To measure the performance of the clustering algorithm, we use three different scoring criteria:

* Silhouette Score
* Calinski Harabasz Score
* Davies Bouldin Score

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 2 clusters | 3 clusters | 6 clusters | 10 clusters |
| Silhouette | 0.5344 | 0.4410 | 0.3800 | 0.3754 |
| Calinski Harabasz | 1345.0079 | 1275.1480 | 1206.9621 | 1235.9940 |
| Davies Bouldin | 0.6339 | 0.8983 | 0.8768 | 0.8192 |

As I pointed out previously, using the dendrogram as a reference, the data points are spreaded out uniformly, so applying clustering on this will not give a clear output. This is why grouping the data points into 2 clusters will have a better result (on all three scoring criteria) than making 3, 6, 10 or more clusters.

References

1. Towards Data Science - Breaking down the agglomerative clustering process <https://towardsdatascience.com/breaking-down-the-agglomerative-clustering-process-1c367f74c7c2>
2. Towards Data Science - Hierarchical Agglomerative Clustering Algorithm Example In Python <https://towardsdatascience.com/machine-learning-algorithms-part-12-hierarchical-agglomerative-clustering-example-in-python-1e18e0075019>
3. Towards Data Science - T-distributed Stochastic Neighbor Embedding(t-SNE) <https://towardsdatascience.com/t-distributed-stochastic-neighbor-embedding-t-sne-bb60ff109561>
4. Towards Data Science - Cheat sheet for implementing 7 methods for selecting the optimal number of clusters in Python <https://towardsdatascience.com/cheat-sheet-to-implementing-7-methods-for-selecting-optimal-number-of-clusters-in-python-898241e1d6ad?gi=6b8c6e353201>