horizontal line

**AIL302M - MACHINE LEARNING**

TRAN THANH DUONG

SE160185

Data clustering using K-Means and PCA

**13th July 2022**

[COLAB NOTEBOOK](https://colab.research.google.com/drive/1tjO5Lqs4tufJNDQpIVxhH0XVlyh25k_b?usp=sharing)

# INTRODUCTION

K-Means is one of the most popular methods in the Clustering problem, with basic idea from calculating distances from data points to their closest centroids and updating the centroids. It's easy to implement and the result will always converge to the optimized solution with any initialized random centroids.

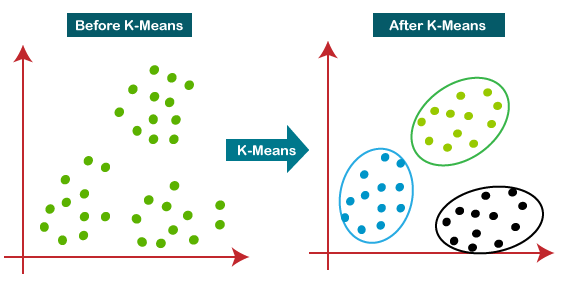
Besides that, PCA is often used to reduce the dimensionality of data, it emphasizes variation and brings out strong patterns in a dataset.

In this assignment report, I will use two of these techniques to find similar patterns from the dataset.

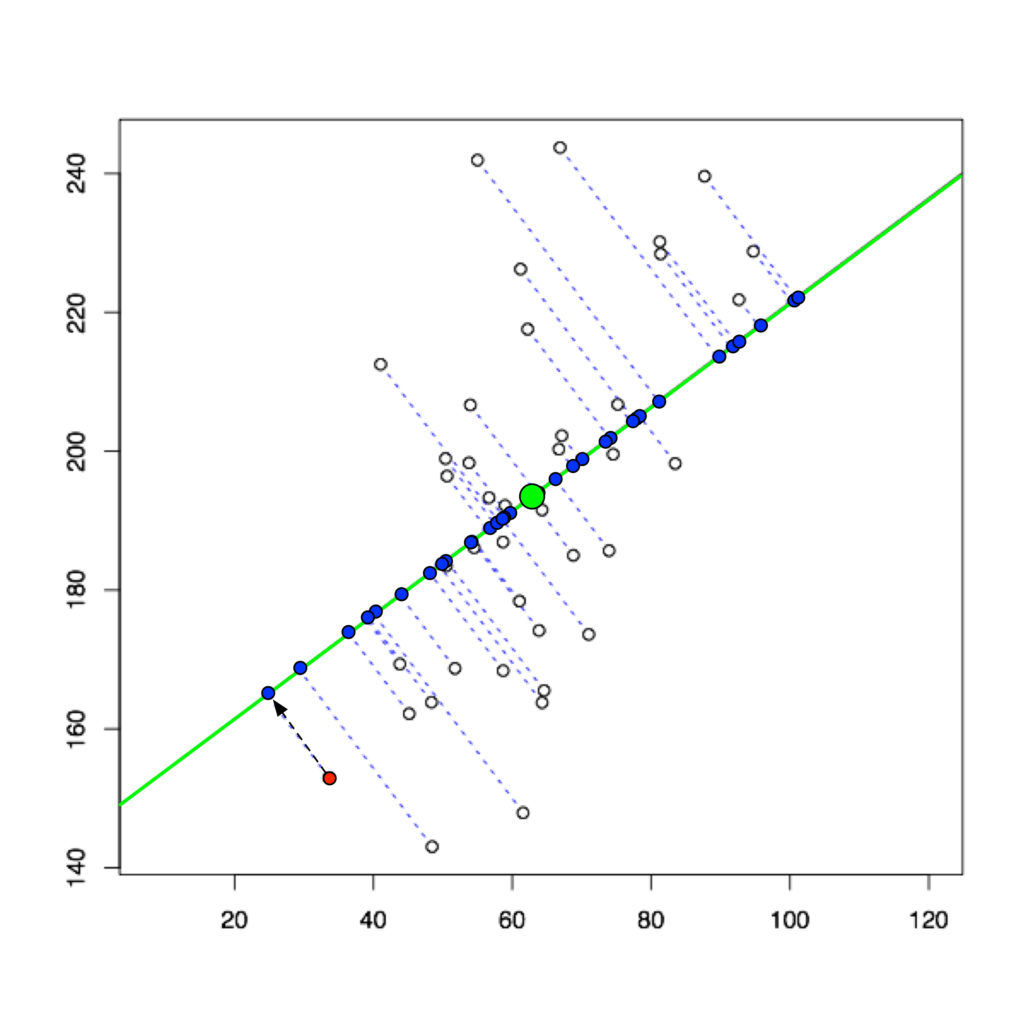
# THE IDEAS BEHIND K-MEANS AND PCA

K-Means method requires predefined "K"-clusters before processing the algorithm. It will randomly initialize "K" points, called **centroids**. After that, applied distance calculating formulas (such as Euclidean distance) to every data point with all points, they will be classified to the cluster of their nearest centroid. Finally, centroids will be updated by calculating the mean of data points in their clusters. Repeat the above steps with a suitable number of iterations until it converges.

The advantages of this method are easy-to-understand and easy to implement in code. The disadvantage is the fixation of the number of clusters and not being capable of handling outliers.



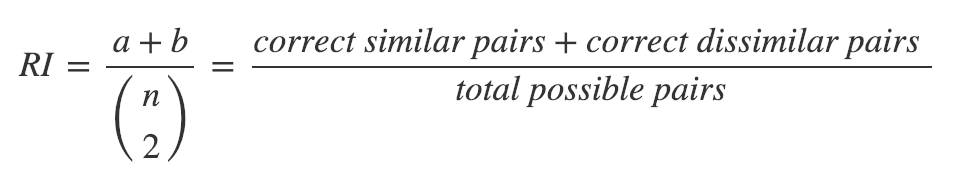
PCA or Principal Components Analysis is used to project data onto a lower-dimensional subspace. A way to find the PCs of data is using eigendecomposition on the covariance of the dataset. The value of eigenvalues will represent the kept ratio information of data when projecting to the corresponding eigenvectors. The new dataset will be created by multiplying the original data with the n-eigenvectors (with n is the number of dimensionalities you want to reduce to).



## METRICS INTRODUCTION

In this assignment, I used two popular metrics in clustering problems:

* **Rand Index Score**: which is a measure of the similarity between two data clusterings.



* **Silhouette Score:** can study the separation distance between the resulting clusters
* Near +1 indicates that the sample is far away from the neighboring clusters.
* A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters
* Negative values indicate that those samples might have been assigned to the wrong cluster.

# METHODS

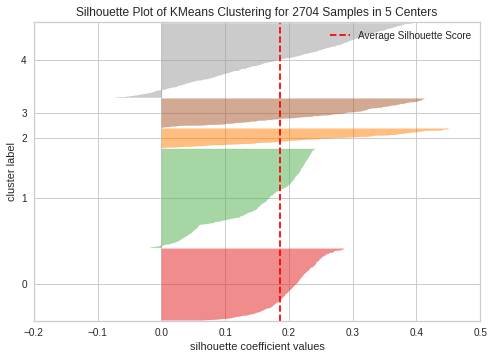
## K-MEANS CLUSTERING

### Number of components = 5

In the very first step, I will try to apply the K-Means algorithm to the original preprocessing dataset with 17 features.

Although it is unsupervised learning, that is, there is no label, I want to verify the accuracy of this method by keeping the labels of the original data and comparing them to the clustering result using the **rand index score**.

I will set **n\_components=5** in K-Means as **5 rating levels** in supervised learning. After training, here is the **Silhouette Plot** of Clustering in 5 Centroids.



#### Table of Results

| **Metrics** | **Value** |
| --- | --- |
| Average Silhouette Score | 0.18674 |
| Training Rand Index Score | 0.61106 |
| Testing Rand Index Score | 0.61202 |

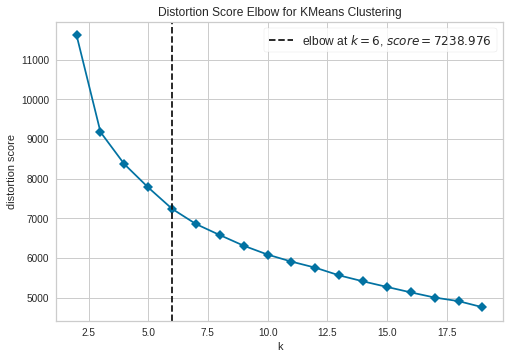
#### Discussion

The results really surprised me when the clustering algorithm gives better performance than the supervised learning algorithm with 61.2% similarity to the test training labels.

### Solutions for choosing the suitable number of clusters - "K"

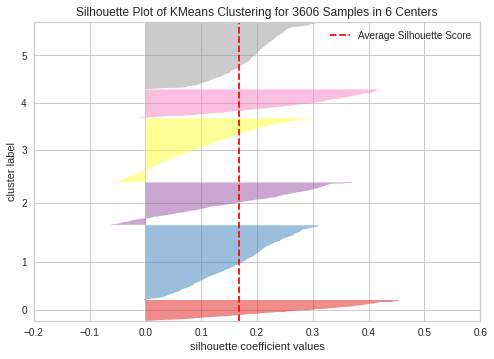
There are usually two common methods for finding the value of "K": Elbow and Silhouette plot.

Here is the result when using the Elbow method to calculate the distortion from 2 clusters to 20 clusters and plot the graph:



As we can see, the elbow at k = 6. After k = 6, adding new clusters does not significantly improve data modeling and the graph tends to decrease linearly.

To check the results, I will use the Silhouette plot to see what the clusters look like:



#### Discussion

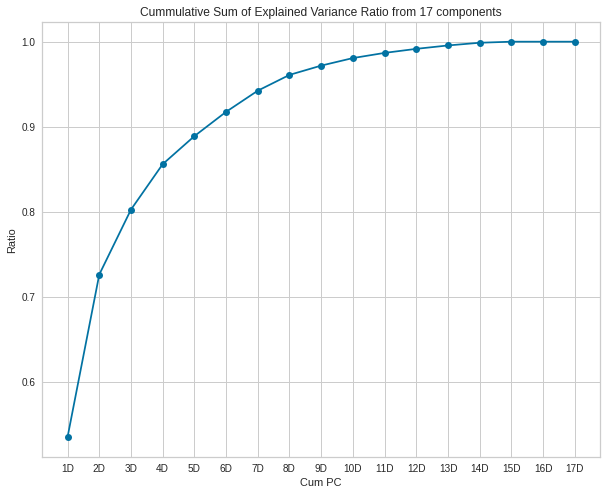
It seems that K=6 is a good pick, although wide fluctuations in the size of the silhouette plots are different significantly but the presence of clusters are above average silhouette scores.

## PRINCIPAL COMPONENTS ANALYSIS (PCA)

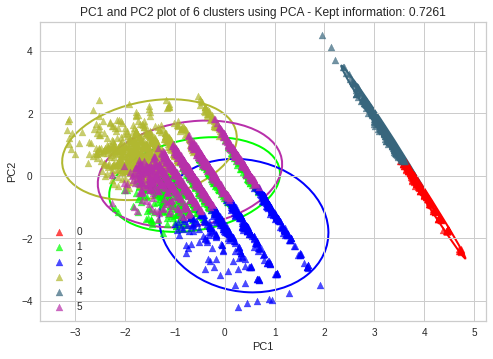
For a better sight of data, we usually plot them to graph visualization. But the original data has 17 features as 17 dimensions, which is not feasible to plot. So we need to have a solution for this, and PCA can solve that problem by reducing dimensionality to 2D or 3D. This method has a disadvantage in that it suffers from data loss when dimensionality reduction.

Further than reducing dimensionality, we want to cluster data on the base of PCA and compare it with K-Means to see what it looks like.

The following plot is the cumulative sum of explained variance ratio from 17 components, the ratio represents the information that can be kept after reducing.



The following clustering graph in 2D and 3D is inherited from the value K=6 of the above K-Means algorithms. The kept information of 2D is 72.6%, and of 3D is 80.3%.



# 

#### Discussion

As we can see in the 2D plot, some clusters are too close and nearly overlap each other. In a 3D plot, with 5% more kept information, we can clearly see the separation between clusters. We can conclude that if two clusters overlap in the 2D or 3D space, it does not imply that they overlap in the higher or original feature space. It gives us information about how close the data points are by looking at the magnitude and intersection of the ellipses. Some clusters are really far from others, causing an imbalance in the amount and coordinates of data points.

# CONCLUSION

To summarize, in part 1, we got unexpected results when performing data clustering and comparing the similarity of the cluster label with the original label with an accuracy of up to >61%, better than many "fancy" supervised learning algorithms that we did in previous assignments. In part 2 using PCA, we better understand how data looks like and the uneven distribution between clusters.