PERSONALITY PREDICTION USING K-MEANS

-VIKAS VELMURUGAN

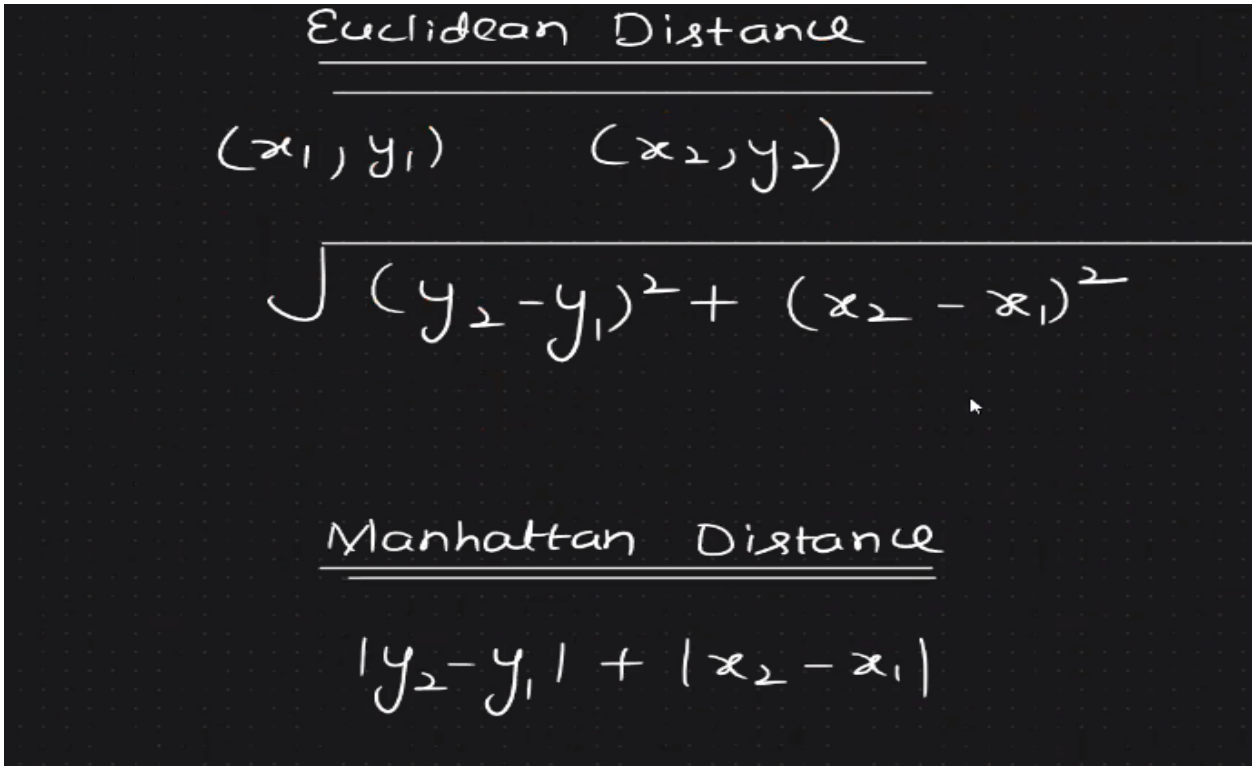
INTRODUCTION

- THE TOP 5 PERSONALITY TRAITS A HUMAN BEING HAS:
  1. **O- Openness** - Willing to explore.
  2. **C - conscientiousness** - Very disciplined.
  3 . **E- Extrovert** - Outgoing
  4. **A - Agreeable** - Highly Co-operative
  5. **N - Neuroticism -** Emotionally unstable/ depressed/ weak mindset.

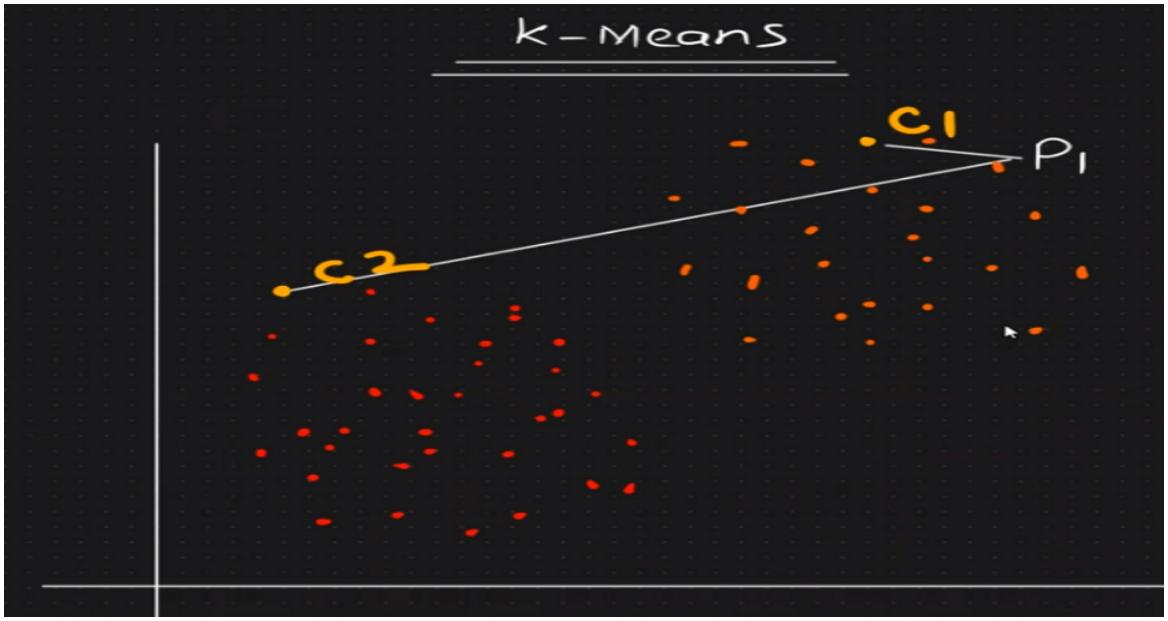
- AIM - To predict the personality of a person by grouping our data into the above 5 categories.
- Basically we need to form 5 clusters, each cluster having a type of trait.
- Clustering is an **Unsupervised Machine Learning** technique.

K-MEANS CLUSTERING

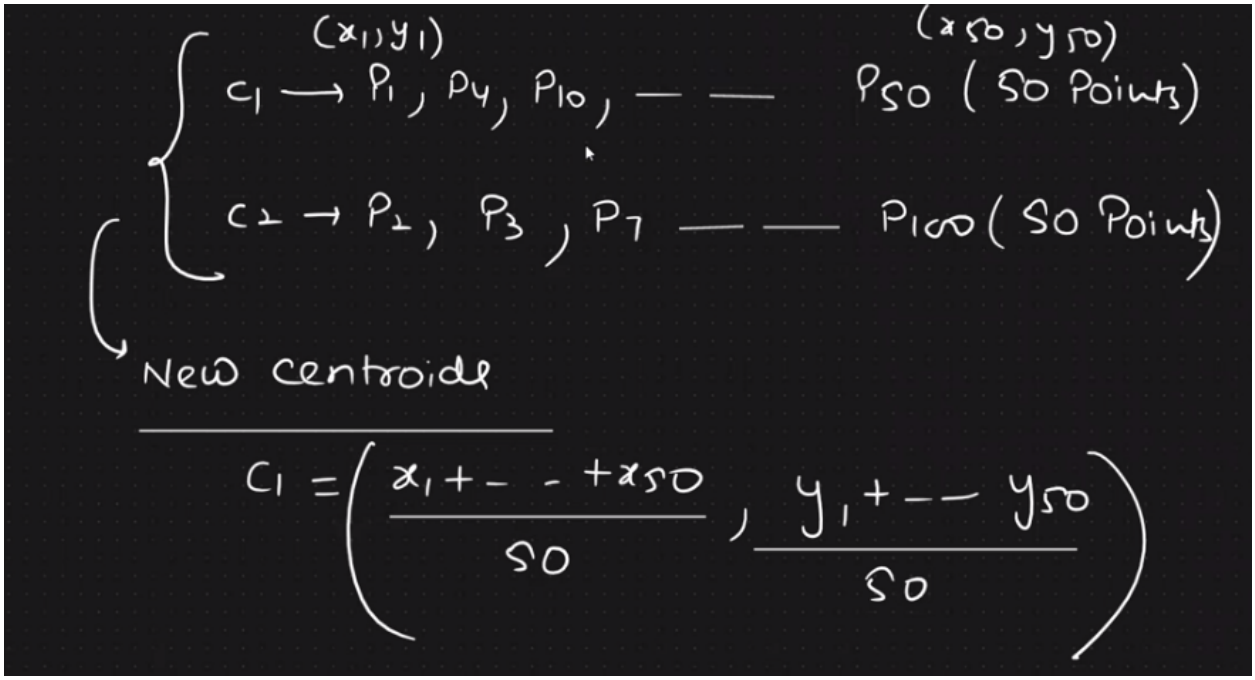
- Choose the number of clusters, say n= 2.
- Based on the n value, n centroids are initialized randomly.
- Distance from random points in the dataset and the centroid is calculated by various distance metrics, usually it is calculated by **Euclidean Distance.**



- Say C1 and C2 are the centroids and P1 is a random data point chosen.
- Distance between C1 and P1 and C2 and P1 is calculated and whichever distance is minimum, then we can say that particular point belongs to that centroid.
- The same process is repeated for all data points in the dataset.
- Say, the dataset has 100 points and C1 has 50 points and C2 has 50 points.



- C1 and C2 are calculated again by taking the average of x and y data points.



- The process is repeated again and new C1 and C2 are calculated.
- **But how long will you repeat this process ? Until C1 and C2 values don't change at all.**
- If you're still not clear, you can visit this link for more information: https://www.gatevidyalay.com/k-means-clustering-algorithm-example/

ELBOW METHOD- Imp Interview Question

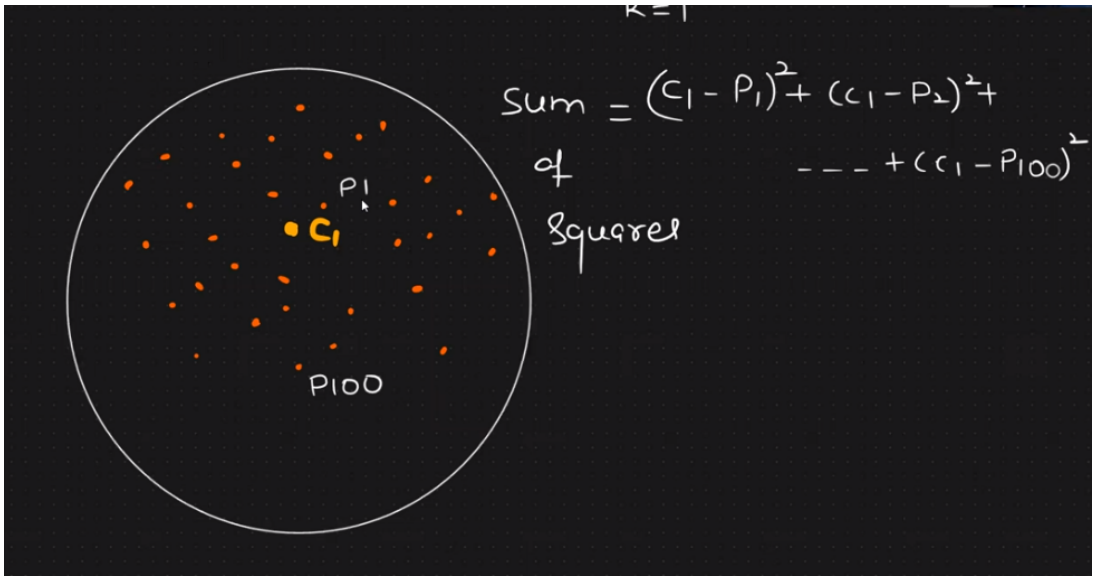
- In this case, we know that we want 5 clusters to categorize our data into 5 personality traits.
- What happens if you're not aware of how many clusters you want? How will you know how many clusters to take?
- To solve this, we have the elbow method:

- **WCSS-** Within cluster sum of squares. Also referred to as inertia sometimes.

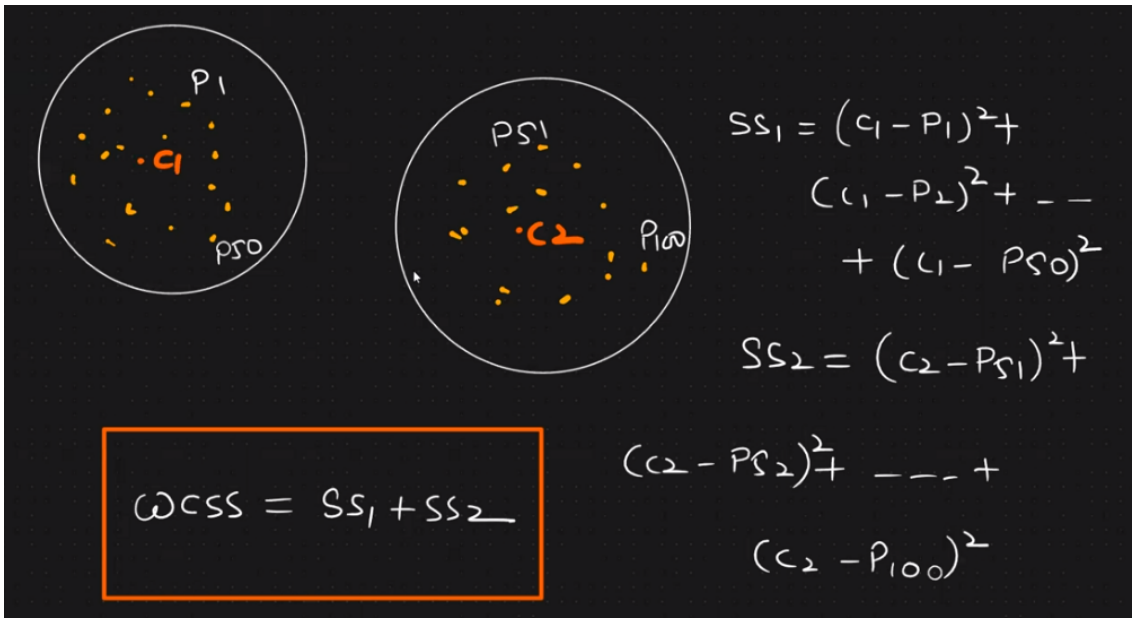
$$WCSS = \sum_{C_k}^{C_n} \left(\sum_{d_i \text{ in } C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

Where,
C is the cluster centroids and d is the data point in each Cluster.

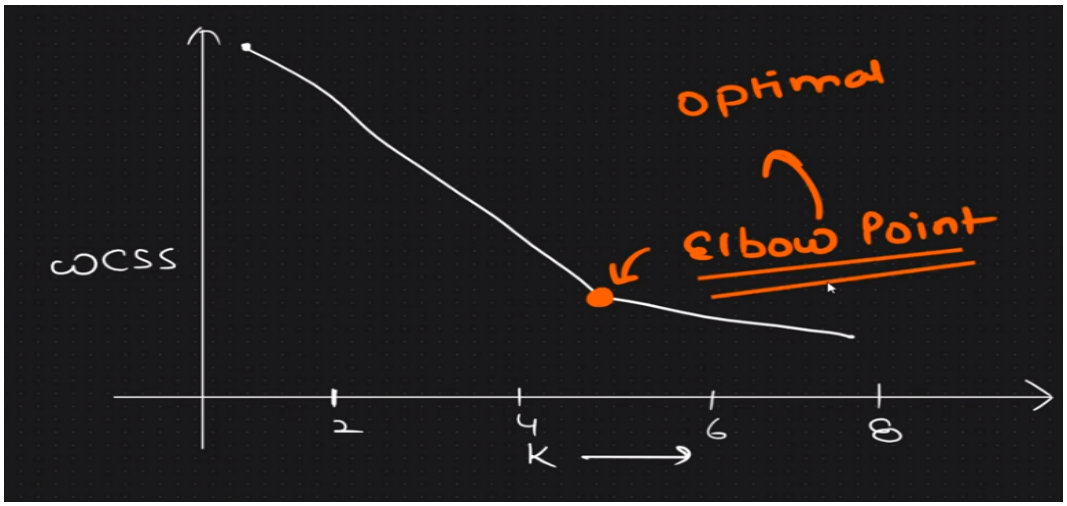
- Initially only one centroid is chosen for all the data points (say 100 data points) and within cluster sum of squares is calculated.



- Next for two clusters : Within cluster sum of squares is calculated again



- When no. of clusters size reaches the total number of data points i.e., cluster size, k = no. of data points, n. Then, WCSS will be zero as all points will become clusters individually and the distance will get canceled out.
- **WCSS will be maximum when k=1 and 0 when k=n. So we can say that WCSS keeps decreasing as K value increases and at a certain value of k, the decrease becomes steady. This forms an elbow point and can be chosen as an optimal value of k.**



DRAWBACKS OF K-MEANS CLUSTERING - IMP INTERVIEW QUESTION

- During random initialisation of centroids, there is a tendency the algorithm might initialize the centroids close to each other and centroids can end up in the same cluster. This makes the clustering very inefficient.
- To solve this, we have **K-Means++** which assures us that the centroids won't be initialized close to each other which helps us get good clusters. The working of the algorithm is the same as the K-means algorithm.
- By default, K-means++ is used.

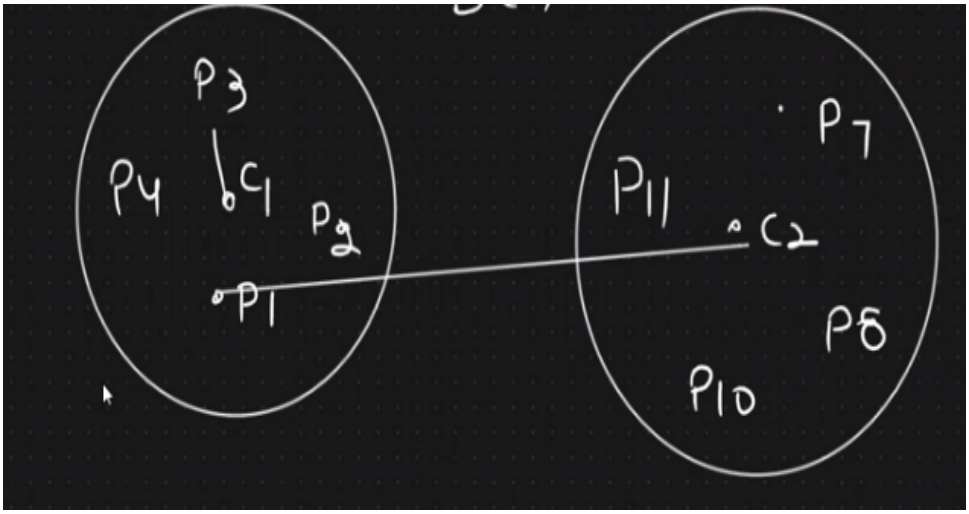
Silhouette Score

For supervised learning, we have many metrics where we can judge the model's performance. But what happens when it comes to unsupervised learning? How do we know that we have good clusters with us?

- For that, we use SilHouette Score.

- b(i) is the intercluster distance(distance of point of centroid of one cluster with data points of another cluster), a(i) is the intracluster distance(distance from centroid with the data points of the same cluster).

- **We can say that , if the clustering is good b(i) should be max cos intercluster distance is high.**
- **Silhouette score ranges from -1 to 1. -1 being bad and 1 being ideal.**



a(i) $\qquad\qquad$ b(i) >> a(i)

$$\text{Score} = \frac{b(i) - a(i)}{\max(b(i), a(i))}$$

CODE IMPLEMENTATION

Dataset link: https://www.kaggle.com/datasets/tunguz/big-five-personality-test