ML Assignment 4

Name - Himang Chandra Garg Roll no. - 2022214

Section A (Theoretical)

Machine Learning
Assignment - 4

Name - Kimang Chandra Garg Roll no. - 2022214

Section A (Theoritical)

Ans 1: (a) given, ① input image dimensions = $M \times N$
② with P channels
③ Kernal size = $K \times K$
④ stride = 1
⑤ padding = 0

so, output height = $\left\lfloor \frac{M-K}{s} \right\rfloor + 1 = M-K+1$.
output width = $\left\lfloor \frac{N-K}{s} \right\rfloor + 1 = N-K+1$.

Ans = $(M-K+1) \times (N-K+1)$.

(b) In each Kernel = $K \times K$ multiplications
input channels = P

① Total multiplications = PK^2

② Total Additions = $P(K^2) - 1$

Ans = $PK^2 + P(K^2) - 1$

Ans = $2PK^2 - 1$

2022219

(c) ① given, Q Kernels
size = K^2

Total no. of operations = Q × operations for 1 pixel × no. of pixels (dimensions)

from b part → $= Q(2PK^2 - 1)\underbrace{(M-K+1)(N-K+1)}_{\text{from (a) part}}$

Ans = $O(QPK^2MN)$.

② given, $\min(M, N) >> K$

so, the term with K (ie K^2 will disappear).

Ans = $O(QPMN)$.

20222214

(b) Assignment Step :-

In this step of the K-means algorithm, we make clusters based on the euclidean distance. We assign each point in the dataset to its closest centroid. Thus ensuring that each cluster contains all the data points closer to its centroid. It based upon the euclidean distance which is calculated as follows :—

$$= \|x_i - m_i\|^2 \quad (\text{norm})$$

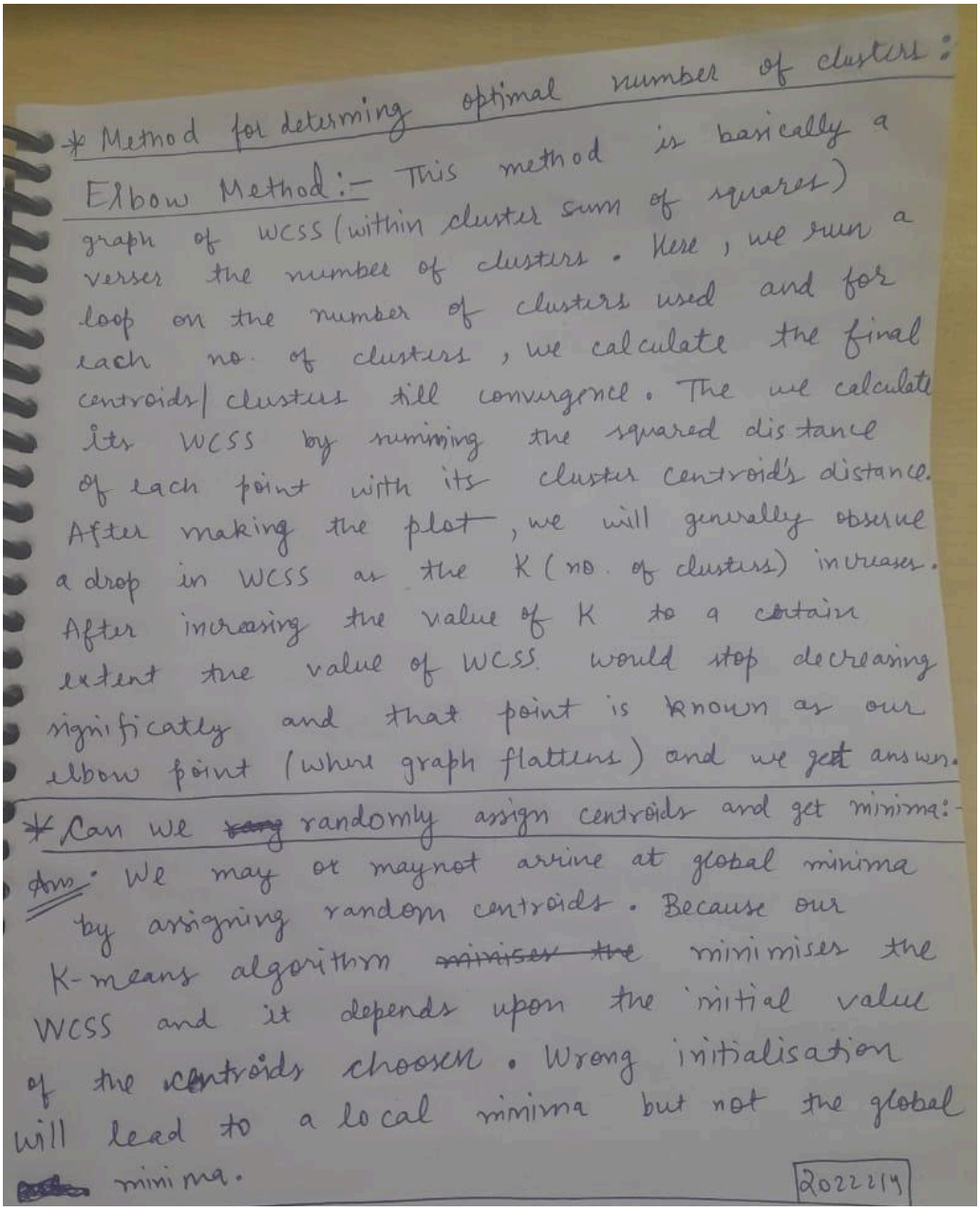
Where, x_i is the data point
and, m_i is the centroid.

Update Step :-

During this step of the K-means algorithm, we update the location of our centroids based upon the clusters formed during assignment step. Thus, for each new cluster formed, the new centroid is calculated as the average of all the points lying inside that cluster.

$$m_x = \frac{1}{n} \sum x_i$$
$$m_y = \frac{1}{n} \sum y_i$$
$$\vdots$$

} for n dimensional data.



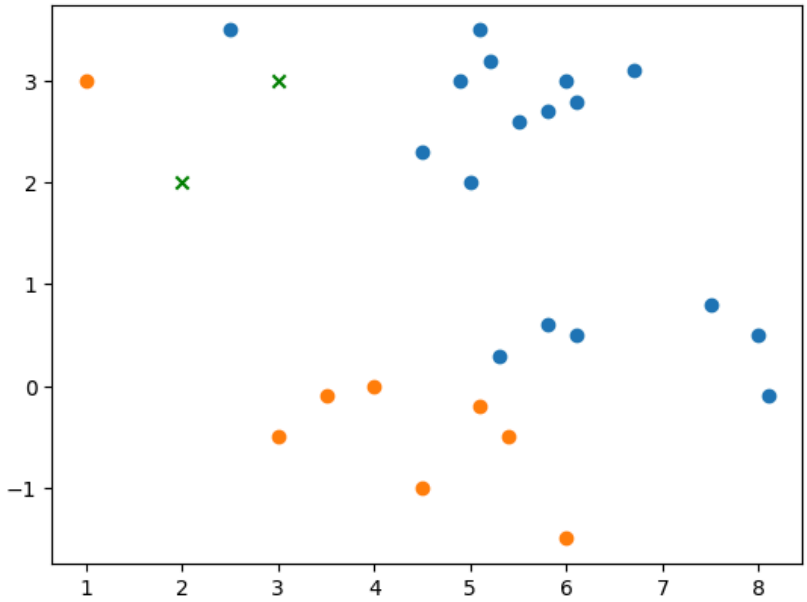
Section B (Scratch Implementation)

(a) Python notebook

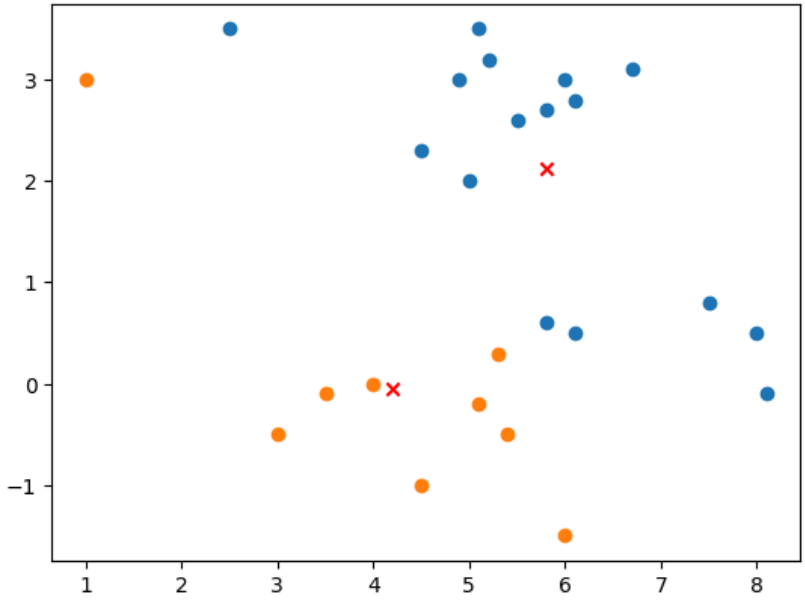
(b) The final value of the centroids:

```
final centroids: [array([5.8  , 2.125]), array([ 4.2       , -0.05555556])]
```

Two clusters at the start:



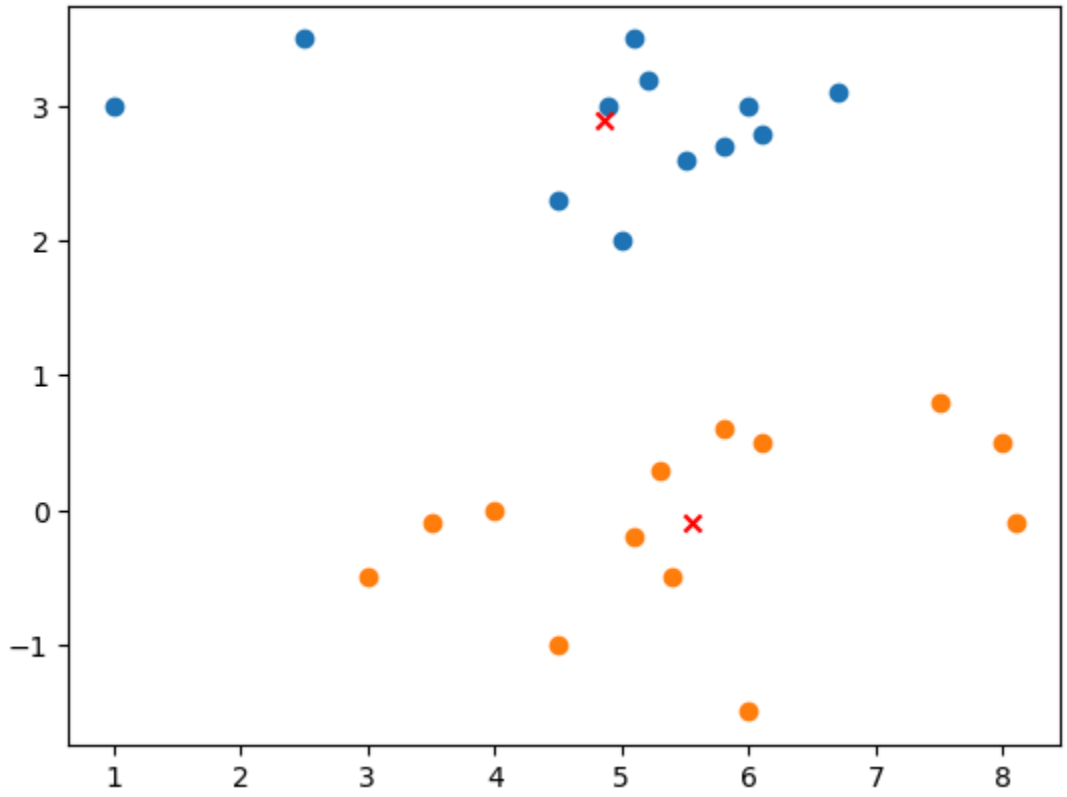
Two clusters at the end:



(d) Final Centroid after random initialization:

```
final centroids: [array([4.85833333, 2.89166667]), array([ 5.56153846, -0.09230769])]
```

Clusters:

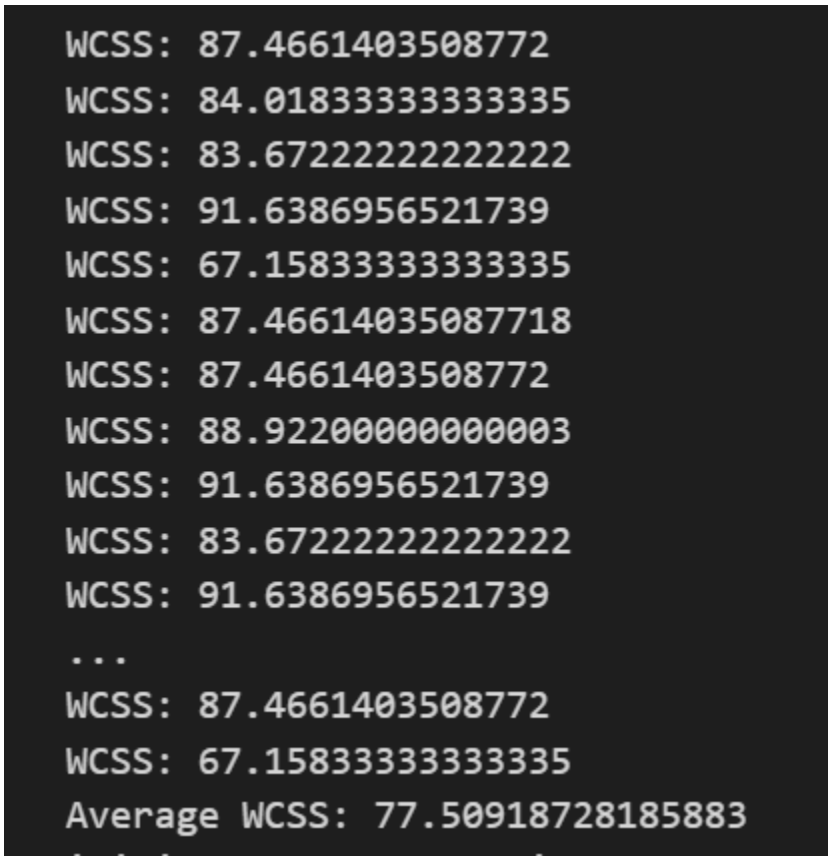


Comparison:

WCSS using the given initial centroids:

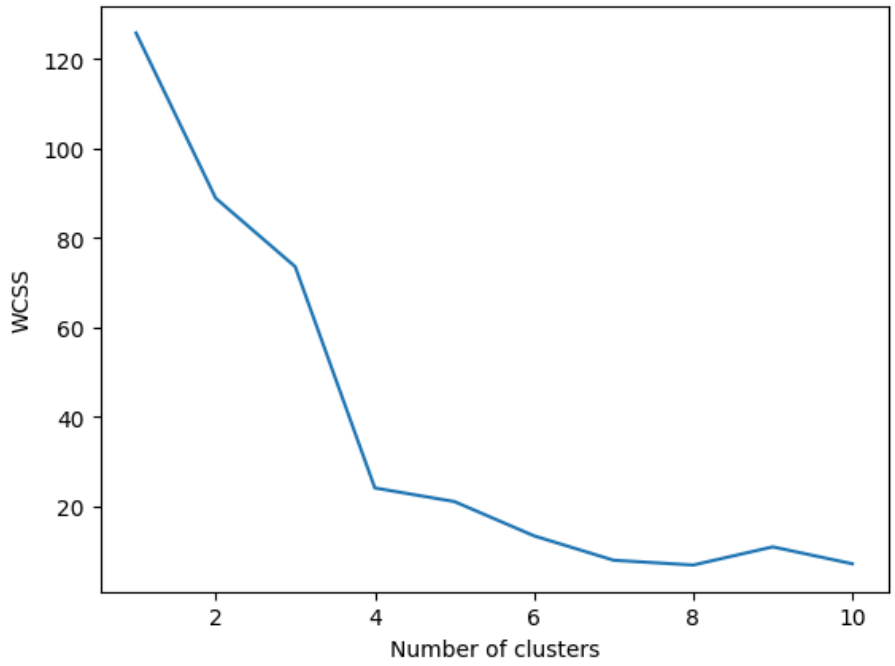
```
WCSS: 83.67222222222222
```

Average WCSS after doing random initialization for 100 times:



As we can observe from the data, on average, random initialization is better than the initial centroids given.

(d) Plotted the graph for the Elbow method(using random initialization):

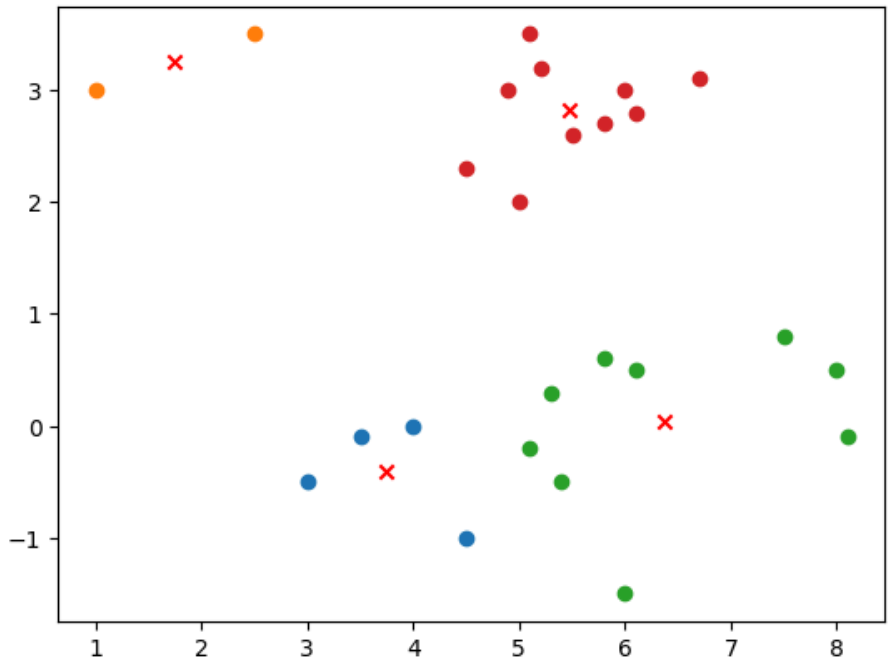


After plotting the graph multiple times, I observed that the optimal number of clusters was 4. Hence, I concluded that the elbow point is 4.

After fixing M=4, I again performed the K means clustering but using 4 clusters this time:
Final centroids:

```
final centroids: [array([ 3.75, -0.4 ]), array([1.75, 3.25]), array([6.36666667, 0.04444444]), array([5.48, 2.82])]
```

Plot:

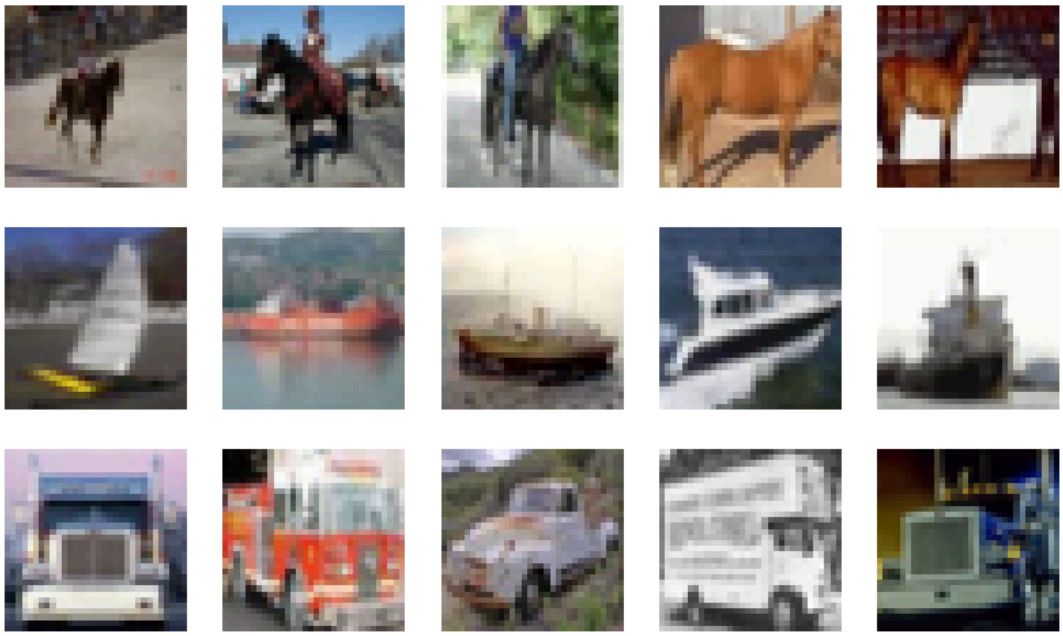

3
2
1
0
−1
1
2
3
4
5
6
7
8

Section C (Algorithm implementation using packages)

(a) Python notebook

(b) Visualization:

Validation Dataset Visualization



Training Dataset Visualization