

ML Assignment 4

Name - Himang Chandra Garg

Roll no. - 2022214

Section A (Theoretical)

Machine Learning
Assignment-4

Name - Himang Chandra Garg

Roll no. - 2022214

Section A (Theoretical)

Ans 1: (a) given, ① input image dimensions = $M \times N$

② with P channels

③ kernel size = $K \times K$

④ stride = 1

⑤ padding = 0

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

output width = $\left\lfloor \frac{N-K}{s} \right\rfloor + 1 = \frac{N-K+1}{s}$

$$\underline{\underline{\text{Ans}}} = (M-K+1) \times (N-K+1)$$

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

① Total multiplications = PK^2

② Total Additions = $P(K^2 - 1)$

$$\underline{\underline{\text{Ans}}} = PK^2 + P(K^2 - 1)$$

$$\boxed{\underline{\underline{\text{Ans}}} = 2PK^2 - P}$$

2022/4

(c) ① given, Q kernels
size = K^2

Total no. of operations = $Q \times$ operations for 1 pixel \times no. of pixels (dimensions)

from b part \downarrow
 $= Q(2PK^2 - 1)(M - K + 1)(N - K + 1)$

Ans = $O(QPK^2MN)$. from (a) part

② given, $\min(M, N) \gg K$

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

Ans = $O(QPMN)$.

202214

(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:-

$$= \|x_i - m_i\|^2 \quad (\text{L2 norm})$$

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.

$$m_{0x} = \frac{1}{n} \sum x_i$$

$$m_{0y} = \frac{1}{n} \sum y_i$$

⋮

} for n dimensional data.

* Method for determining optimal number of clusters :-

Elbow Method:- This method is basically a graph of WCSS (within cluster sum of squares) versus the number of clusters. Here, we run a loop on the number of clusters used and for each no. of clusters, we calculate the final centroids/ clusters till convergence. Then we calculate its WCSS by summing the squared distance of each point with its cluster centroid's distance. After making the plot, we will generally observe a drop in WCSS as the K (no. of clusters) increases. After increasing the value of K to a certain extent the value of WCSS would stop decreasing significantly and that point is known as our elbow point (where graph flattens) and we get answer.

* Can we ~~very~~ randomly assign centroids and get minima:-

Ans:- We may or maynot arrive at global minima by assigning random centroids. Because our K-means algorithm ~~minimise the~~ minimises the WCSS and it depends upon the initial value of the centroids chosen. Wrong initialisation will lead to a local minima but not the global minima.

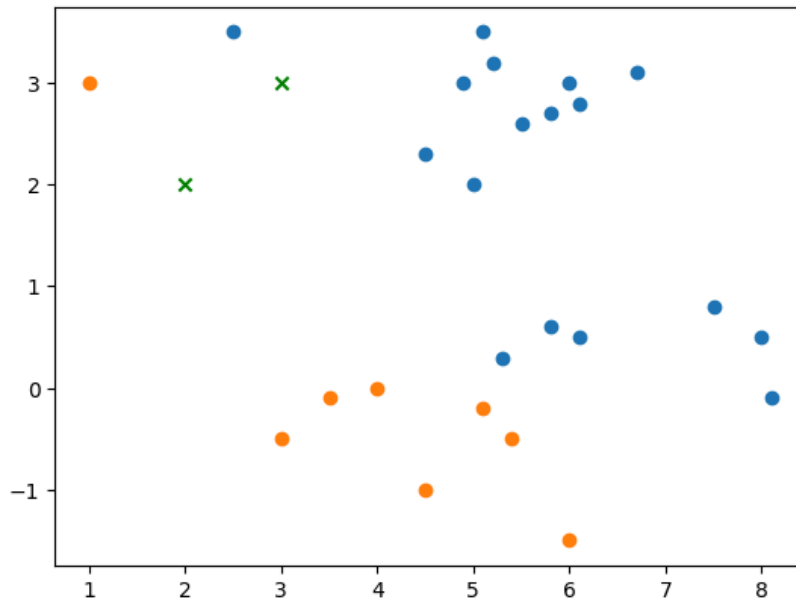
2022219

Section B (Scratch Implementation)

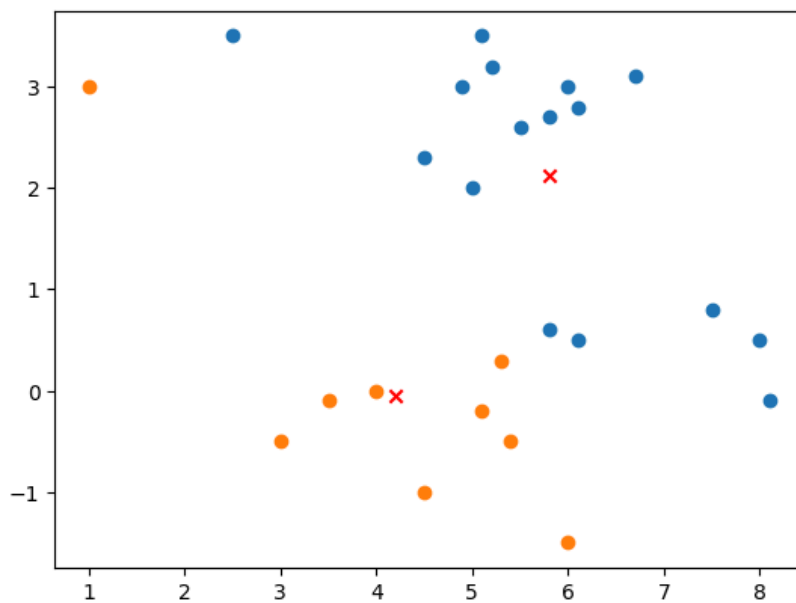
- (a) Python notebook
- (b) The final value of the centroids:

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

Two clusters at the start:



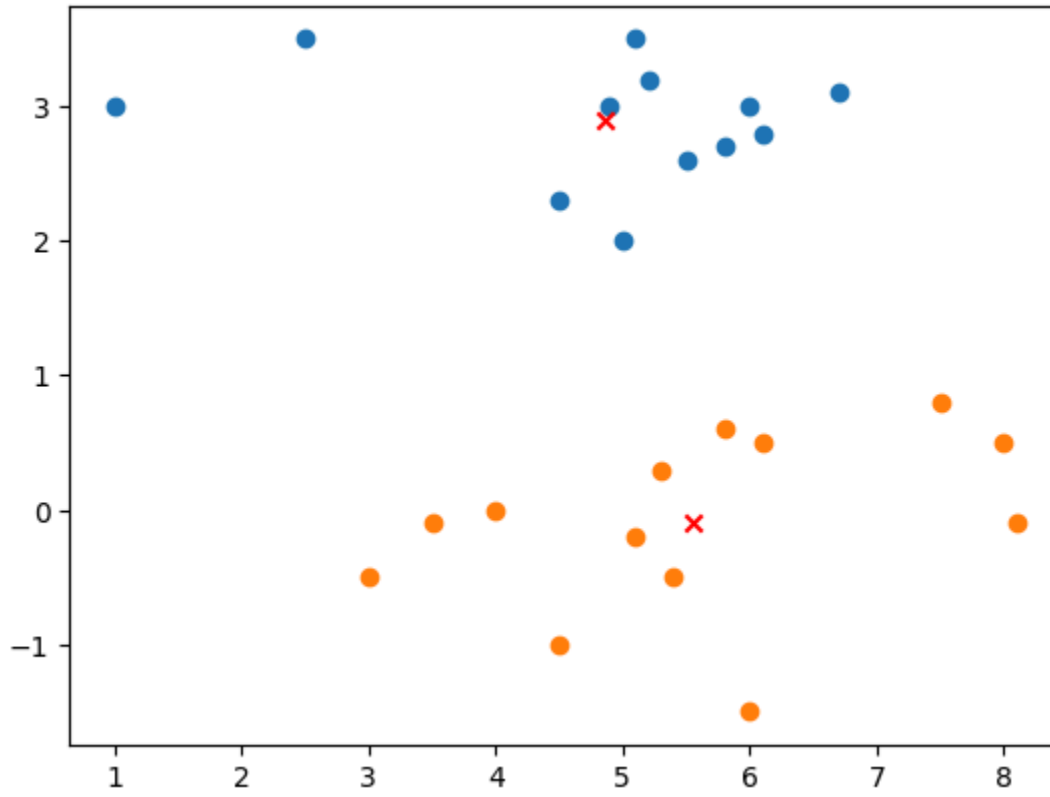
Two clusters at the end:



(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:

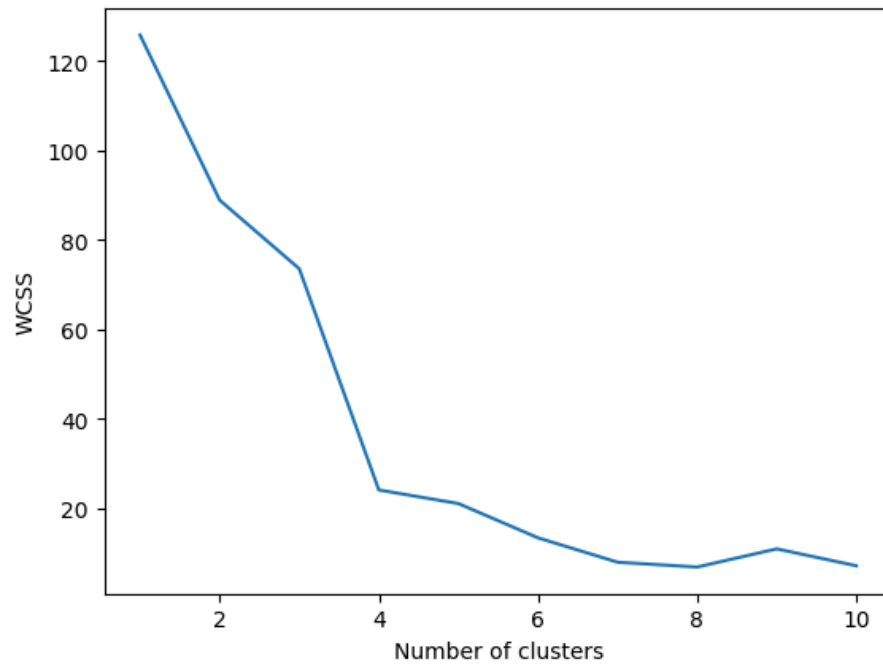
WCSS: 83.67222222222222

Average WCSS after doing random initialization for 100 times:

```
WCSS: 87.4661403508772
WCSS: 84.01833333333335
WCSS: 83.67222222222222
WCSS: 91.6386956521739
WCSS: 67.15833333333335
WCSS: 87.46614035087718
WCSS: 87.4661403508772
WCSS: 88.92200000000003
WCSS: 91.6386956521739
WCSS: 83.67222222222222
WCSS: 91.6386956521739
...
WCSS: 87.4661403508772
WCSS: 67.15833333333335
Average WCSS: 77.50918728185883
```

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):



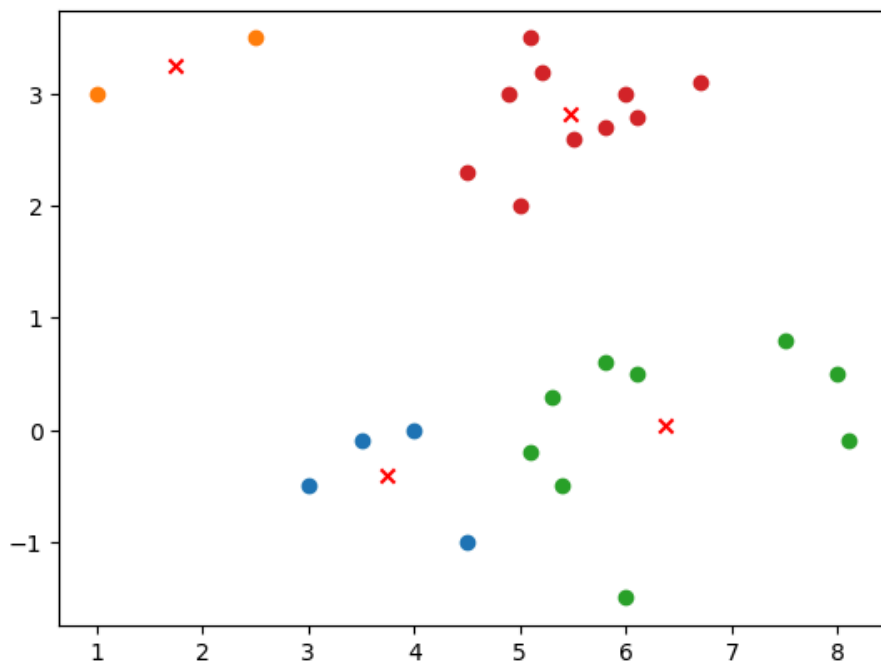
After plotting the graph multiple times, I observed that the optimal number of clusters was 4. Hence, I concluded that the elbow point is 4.

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:

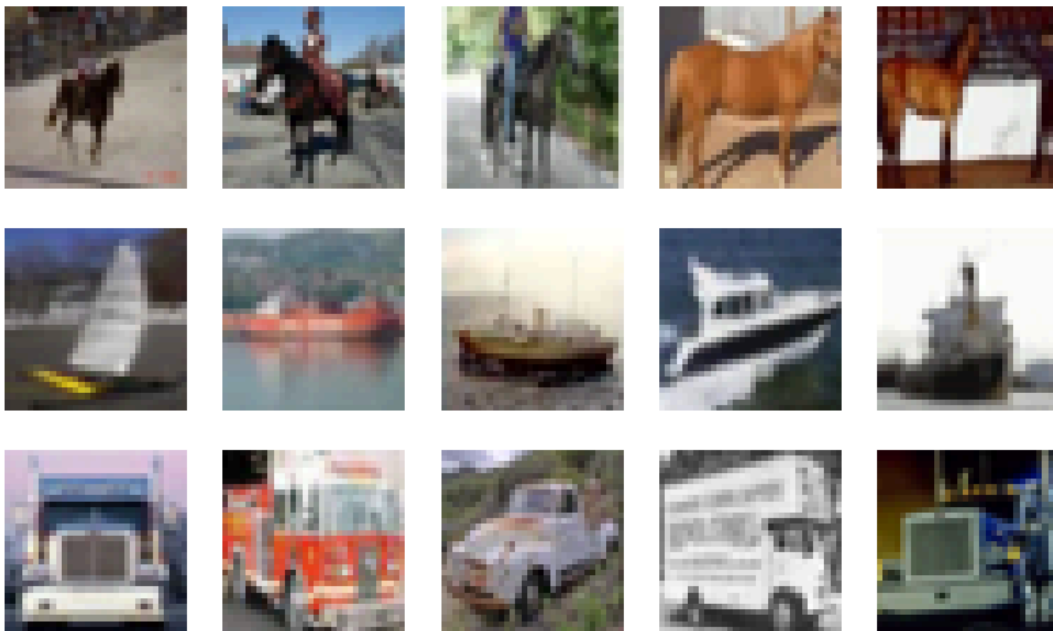


Section C (Algorithm implementation using packages)

(a) Python notebook

(b) Visualization:

Validation Dataset Visualization



Training Dataset Visualization

