K MEANS CLUSTERING

USING PYTHON
DR. ALVIN ANG

CONTENTS

I. Understanding K-Means Clustering = Unsupervised Machine Learning 3

A. in a Nutshell…. 4

B. Step 1: Choose the Number of K (Centroids) 5

C. Step 2: Random Initialization of Centroids 5

D. Step 3: Assigning Cluster Number to each Point 6

E. Step 4: Calculate New Centroids And Reassign New Clusters 7

F. Keep Repeating Step 4 8

G. Elbow Method 9

II. Learning K Means Using Orange 10

A. Download Orange 10

B. Paint Data 10

C. Add On Educational Package 11

D. Observing Interactive K Means 12

III. K Means Using Python (SciKit Learn) 13

A. Import Libraries 13

B. Generate Random Data 14

1. Creating X 14
2. Creating X1 14
3. Replacing X 50th to 100th row 15

C. Plot the Random Data 16

D. Importing K Means 16

E. Centroids 17

F. Labeling 18

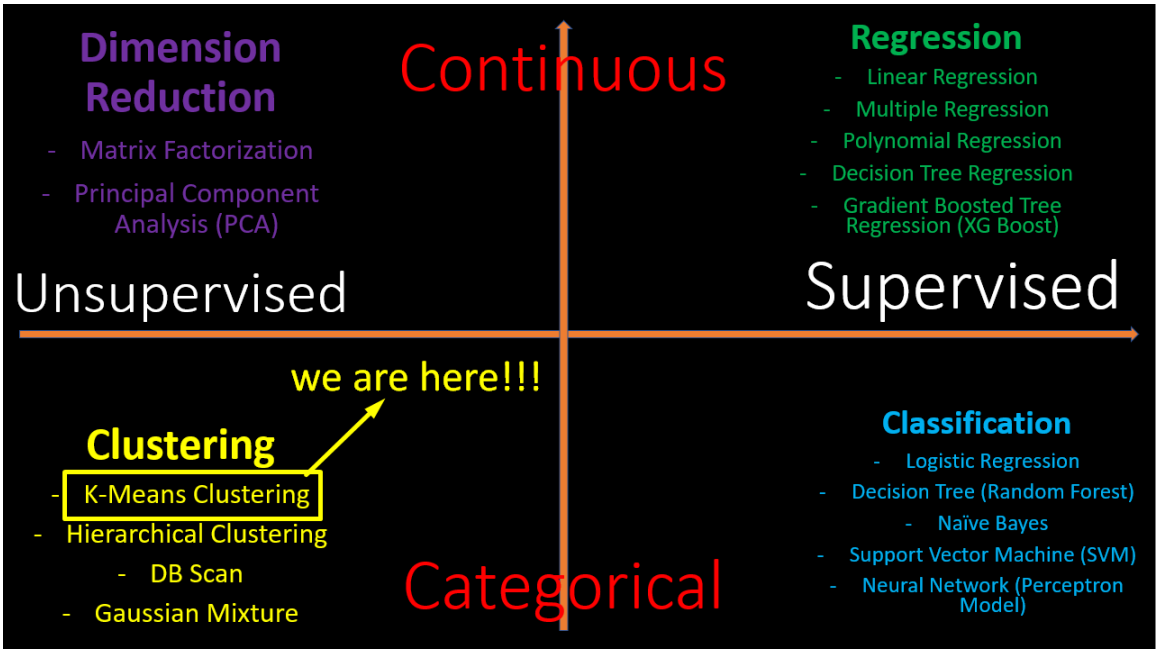
G. Predicting a New Coordinate 18

About Dr. Alvin Ang 19

I. UNDERSTANDING K-MEANS CLUSTERING = UNSUPERVISED MACHINE LEARNING

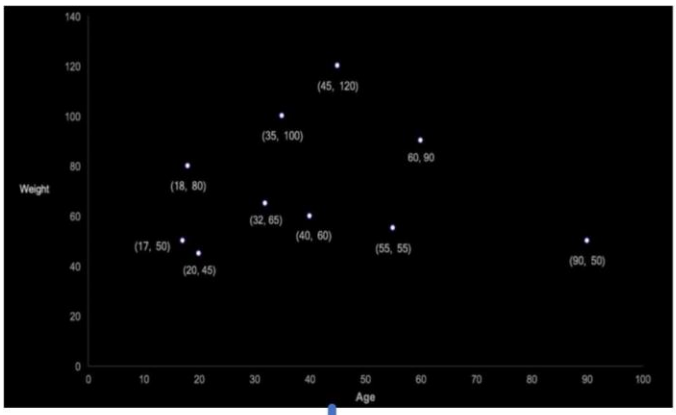
Most of the stuff here are abstracted frrom:

https://www.amazon.com/Machine-Learning-PySpark-Processing-Recommender/dp/1484241304



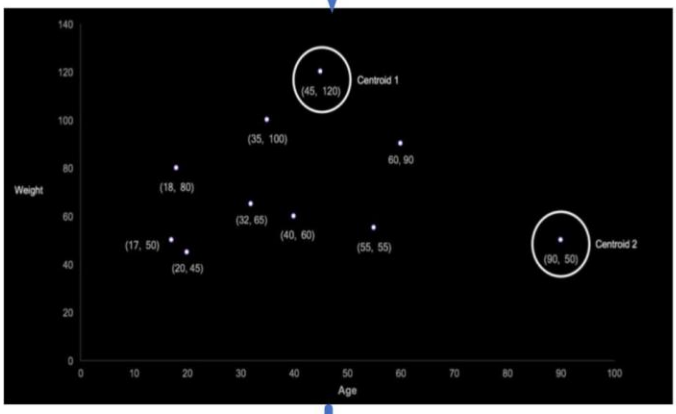
- Above is a table categorizing the different Machine Learning algorithms.
- Objective of K-Means Clustering is to predict a CATEGORY.

A. IN A NUTSHELL….



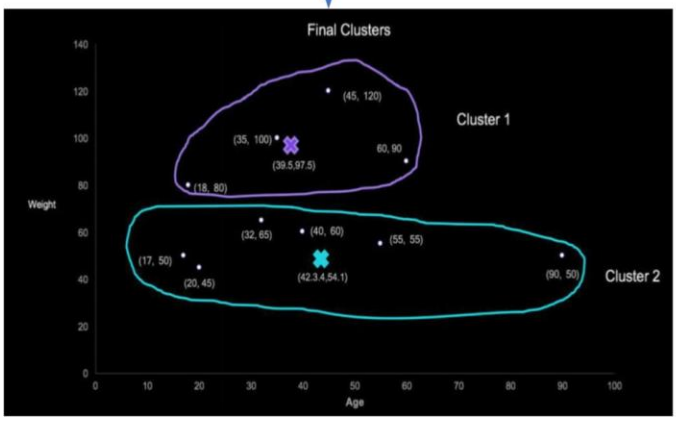
Stage 1: Many Data Points

We want to Cluster them.



Stage 2: Randomly Select Clustering Centroids

We randomly choose 2 clustering starting points (known as Centroids) and start running the K Means algorithm.



Stage 3: Final Labeling of Clusters

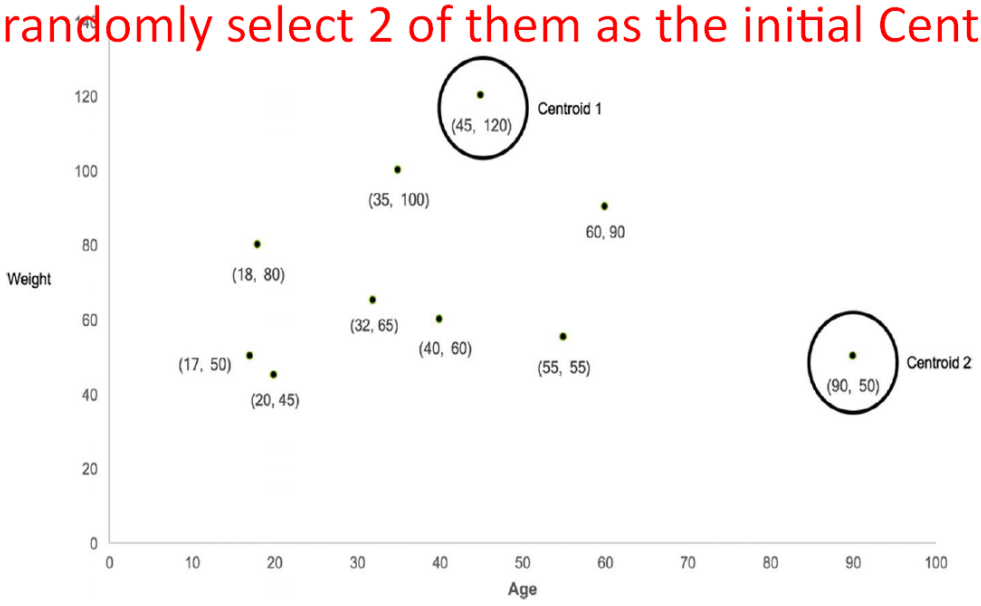
Every data point is labelled either 1 or 2 and the Centroid location coordinate is found.

B. STEP 1: CHOOSE THE NUMBER OF K (CENTROIDS)

- This is a tricky question.
- We either use gut feel / domain business understanding / quantitative methods.
- We shall dwell more in the later section (using the Elbow method) to derive K.
- For now, we just assume 2 Centroids (2 Clusters).

C. STEP 2: RANDOM INITIALIZATION OF CENTROIDS

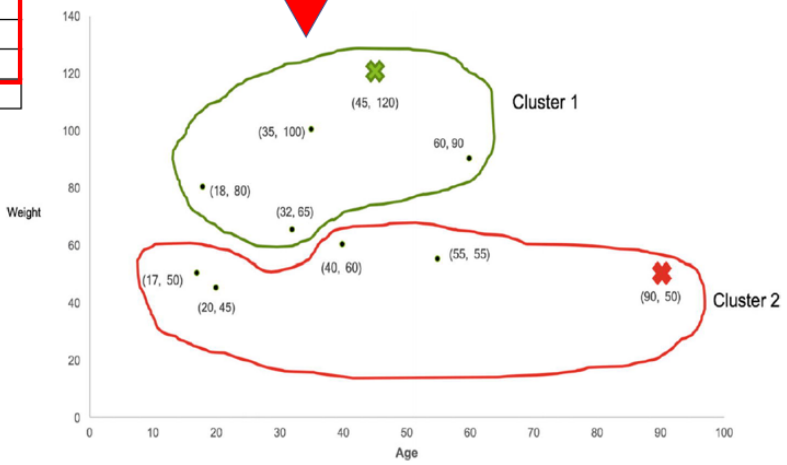
we have some random data points and we randomly select 2 of them as the initial Centroids..



D. STEP 3: ASSIGNING CLUSTER NUMBER TO EACH POINT

User ID	Age	Weight	ED* from Centroid 1	ED* from Centroid 2	Cluster
1	18	80	48	78	1
2	40	60	60	51	2
3	35	100	22	74	1
4	20	45	79	70	2
5	45	120	0	83	1
User ID	Age	Weight	ED* from Centroid 1	ED* from Centroid 2	Cluster
6	32	65	57	60	1
7	17	50	75	73	2
8	55	55	66	35	2
9	60	90	34	50	1
10	90	50	83	0	2
(*Euclidean Distance)					

We label every point to a Cluster 1 or 2 Based on their nearest distance to the Centroid



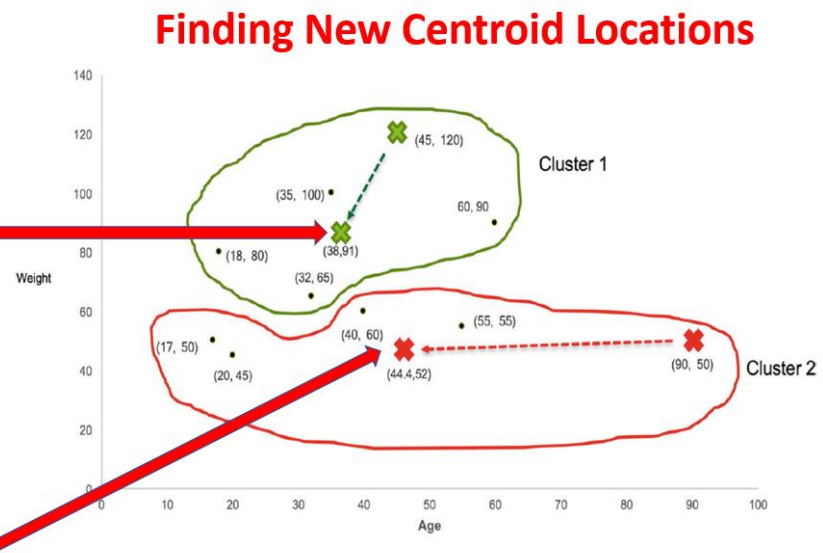
E. STEP 4: CALCULATE NEW CENTROIDS AND REASSIGN NEW CLUSTERS

Initial Cluster 1

User ID	Age	Weight
1	18	80
3	35	100
5	45	120
6	32	65
9	60	90
Mean Value	**38**	**91**

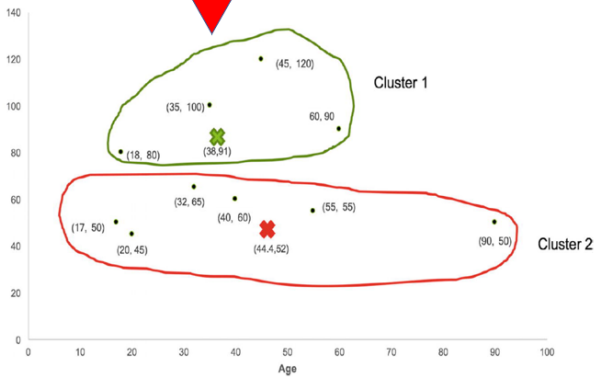
Initial Cluster 2

User ID	Age	Weight
2	40	60
4	20	45
7	17	50
8	55	55
10	90	50
Mean Value	**44.4**	**52**

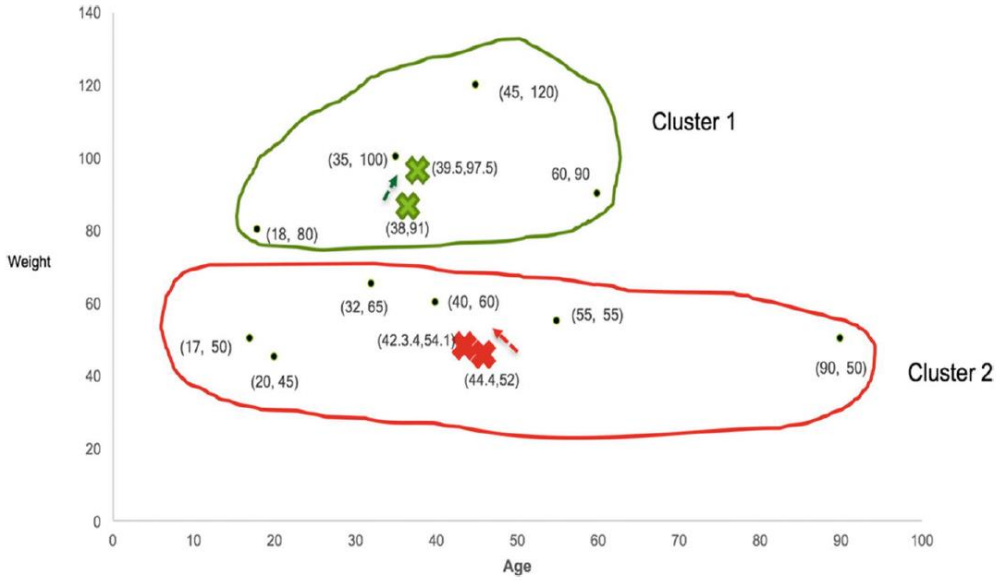


User ID	Age	Weight	ED* from Centroid 1	ED* from Centroid 2	Cluster
1	18	80	23	38	1
2	40	60	31	9	2
3	35	100	9	49	1
4	20	45	49	25	2
5	45	120	30	68	1
6	32	65	27	18	2
7	17	50	46	27	2
8	55	55	40	11	2
9	60	90	22	41	1
10	90	50	66	46	2

We RElabel every point to a Cluster 1 or 2 Based on their nearest distance to the Centroid

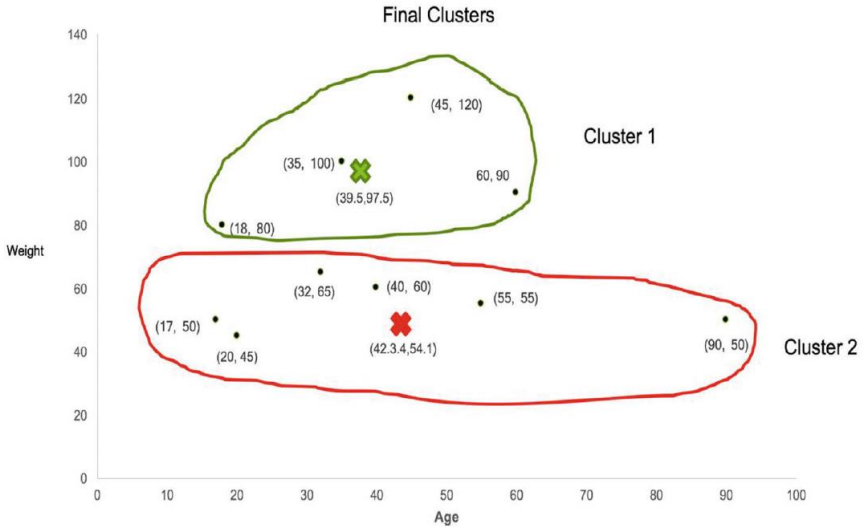


F. KEEP REPEATING STEP 4

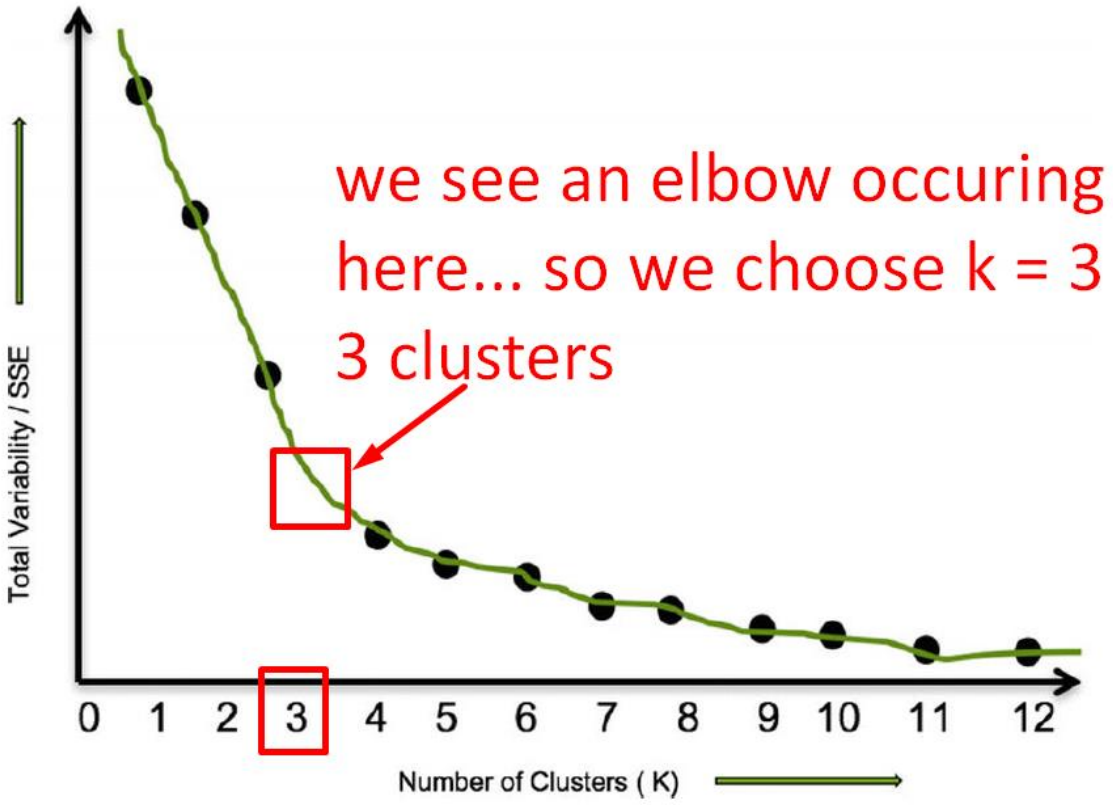


- Every time you repeat Step 4, you will notice the Centroid move incrementally smaller and smaller distance….

- Until you can't move anymore… you have achieved the final centroids….



G. ELBOW METHOD



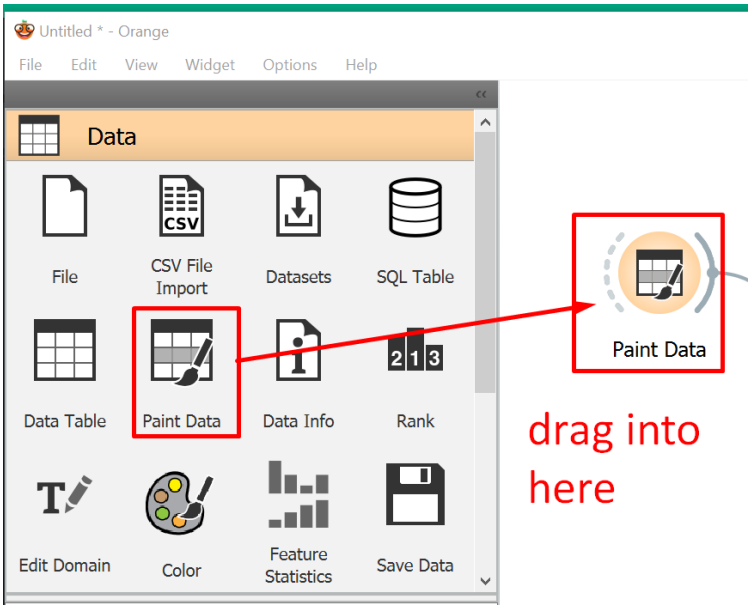
- Elbow Method deals with Step 1: Choosing the number of K (Centroids).
- Total Variability or SSE is a measure of how far the distance is from the Points to the Centroid.
- So let's say we have 12 points and all 12 points are 12 Centroids → SSE = 0 because the distance from each Centroid to each Point = 0 (because it is to itself!)
- Thus, the more the number of clusters, the lesser the variance.
- We choose K = 3 where the Elbow occurs.

II. LEARNING K MEANS USING ORANGE

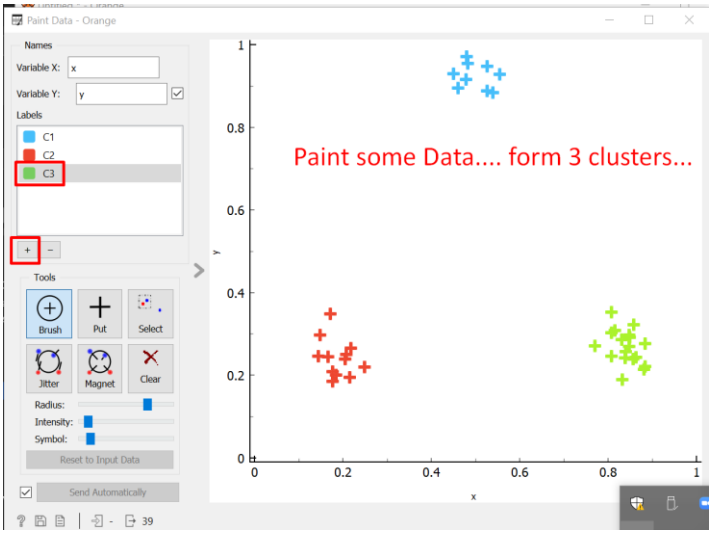
A. DOWNLOAD ORANGE

- Download orange and install from https://orangedatamining.com/download/#windows

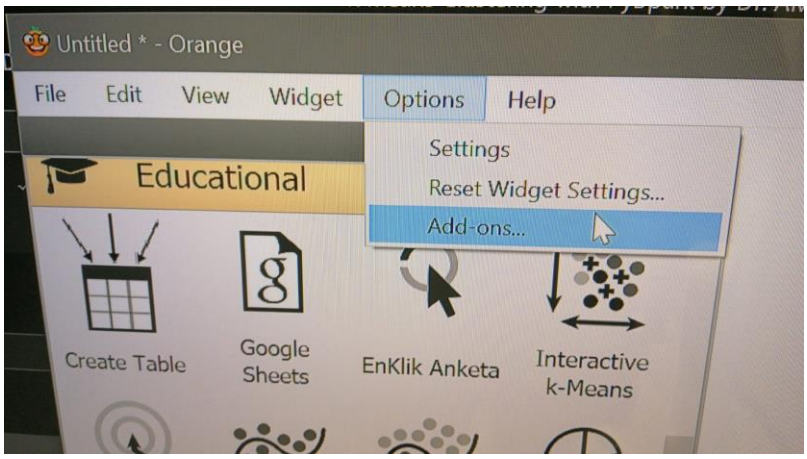
B. PAINT DATA

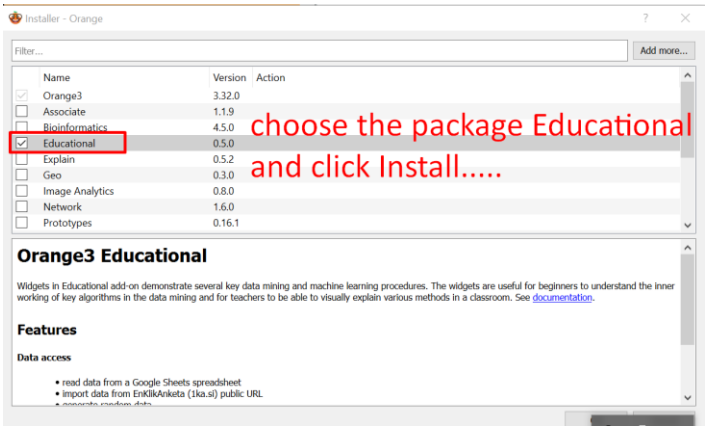


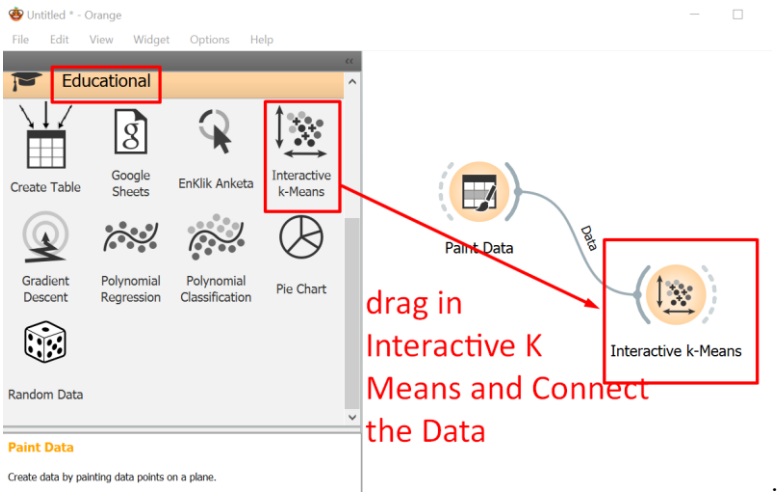
… Double click the icon…



C. ADD ON EDUCATIONAL PACKAGE

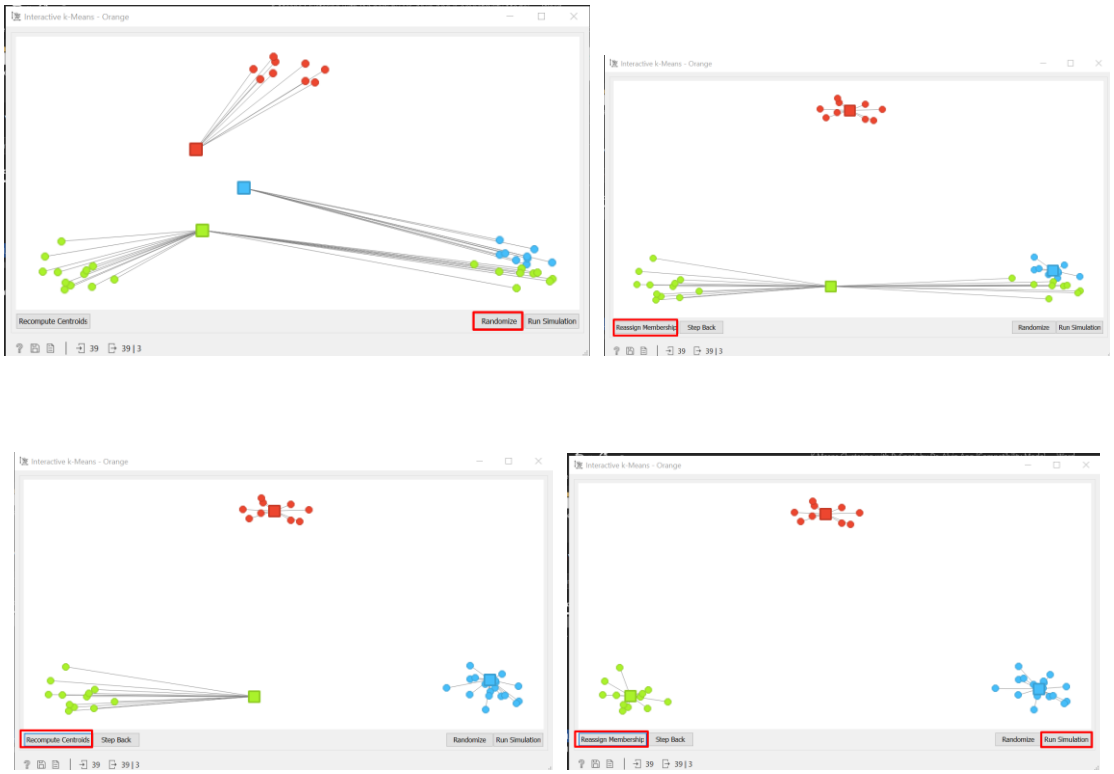






.. Double Click the Icon…

D. OBSERVING INTERACTIVE K MEANS



- In the beginning, you click on Randomize to initialize 3 randomly picked Centroids…
- Step by step, as you click on Recompute Centroids, you slowly see the 3 clusters converging…and the Centroids closing in until it can't move anymore….

III. K MEANS USING PYTHON (SCIKIT LEARN)

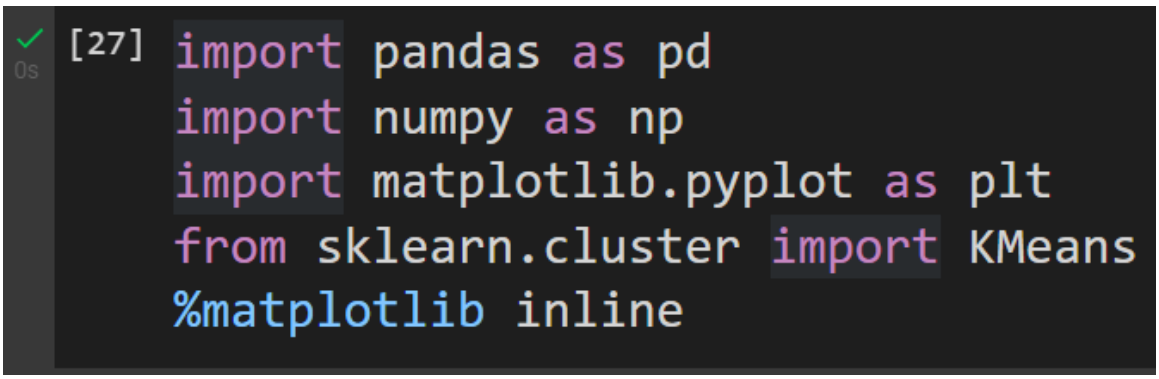
Most of the code is taken from:

https://medium.com/towards-data-science/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Ipynb file here:

https://www.alvinang.sg/s/KMeans_using_Python_by_Dr_Alvin.ipynb

A. IMPORT LIBRARIES



B. GENERATE RANDOM DATA

1. CREATING X

```
# generating some random data in a two-dimensional space
# A total of 100 data points has been generated and divided into two groups, of 50 points each.

X = -2 * np.random.rand(100,2)
X
```

```
array([[-1.07454338e+00, -4.94808659e-01],
       [-7.27578981e-01, -1.44145986e+00],
       [-2.48742904e-01, -2.76016633e-01],
       [-1.97818809e+00, -9.00636156e-01],
       [-9.43871177e-01, -5.90283784e-01],
       [-1.29295121e+00, -1.68408531e+00],
       [-9.65537842e-01, -6.28839774e-05],
       [-2.09497227e-01, -4.34832927e-01],
```

100 random negative pair values created and stored in X

X is now an array with 100 rows and 2 columns

2. CREATING X1

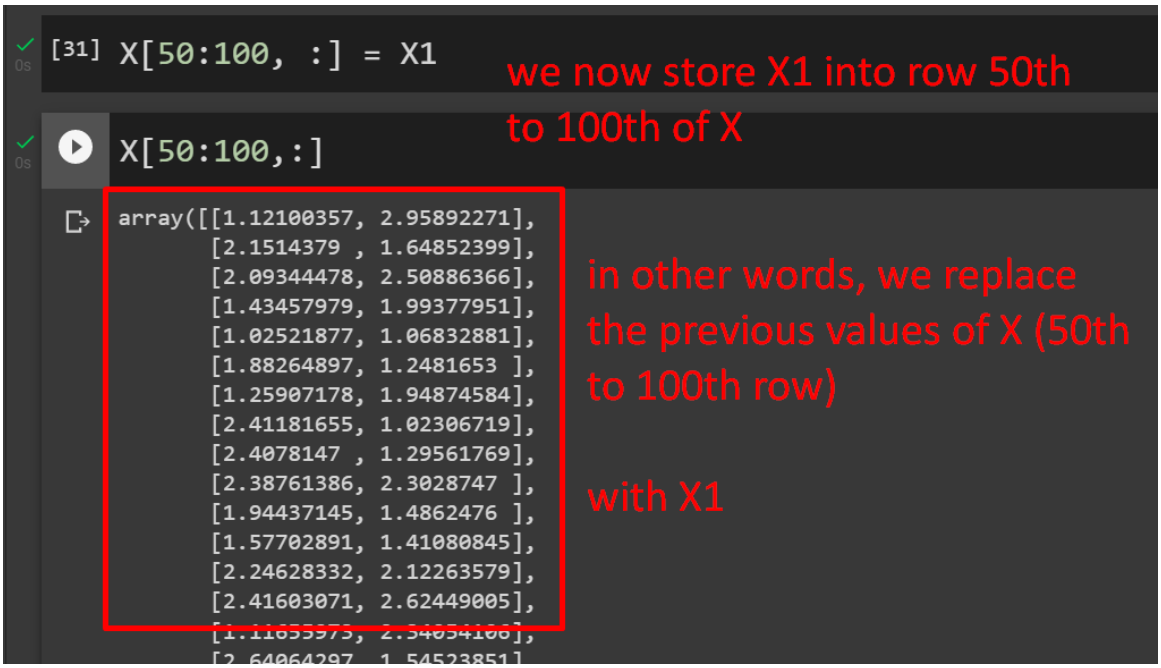
```
X1 = 1 + 2 * np.random.rand(50,2)
X1
```

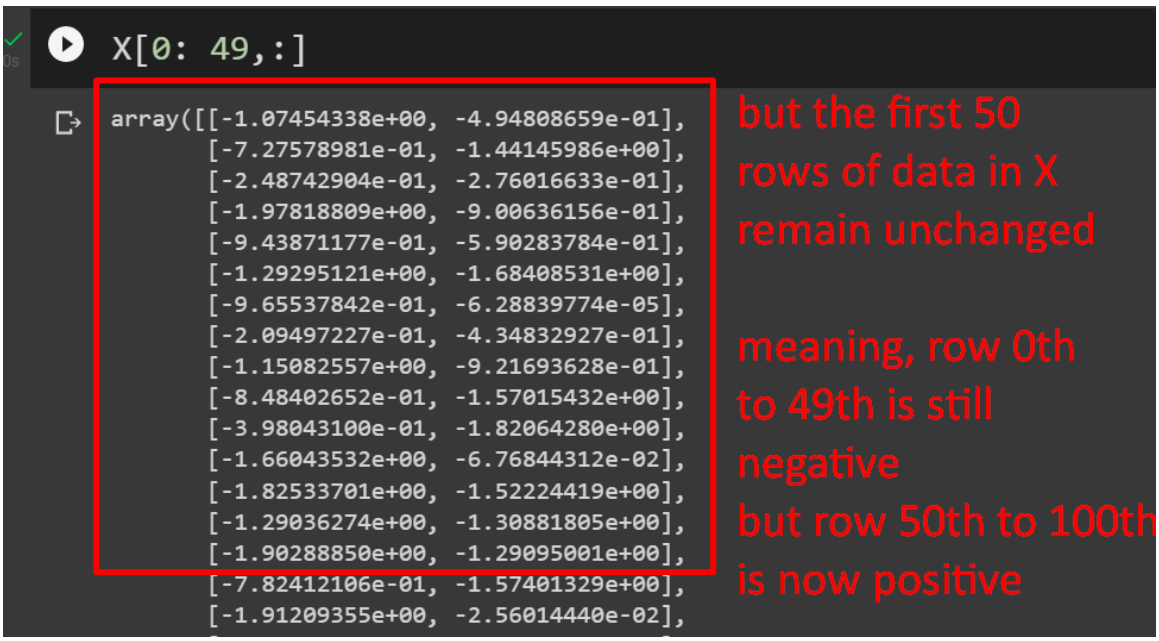
```
array([[1.12100357, 2.95892271],
       [2.1514379 , 1.64852399],
       [2.09344478, 2.50886366],
       [1.43457979, 1.99377951],
       [1.02521877, 1.06832881],
       [1.88264897, 1.2481653 ],
       [1.25907178, 1.94874584],
       [2.41181655, 1.02306719],
       [2.4078147 , 1.29561769],
       [2.38761386, 2.3028747 ],
       [1.94437145, 1.4862476 ],
       [1.57702891, 1.41080845],
       [2.24628332, 2.12263579],
       [2.41603071, 2.62449005],
```

50 positive random paired values have been created and stored in X1

X1 is now an array of 50 rows and 2 columns

3. REPLACING X 50TH TO 100TH ROW





C. PLOT THE RANDOM DATA

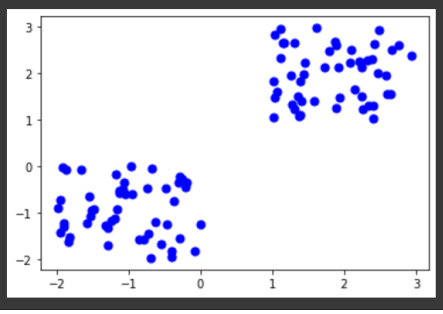
```
plt.scatter(X[ : , 0], X[ :, 1], s = 50, c = 'b')
plt.show()
```

x axis

y axis

size of dots = 50

color = blue



now we see that the first column of X is input to the x scatter plot.

the second column of X is input to the y of the scatter plot.

in other words, all negative coordinates (row 0 to 49) are plotted.
all positive coordinates (row 50 to 100) are plotted.

D. IMPORTING K MEANS

```
[41] from sklearn.cluster import KMeans
     Kmean = KMeans(n_clusters=2)
     Kmean.fit(X)

     KMeans(n_clusters=2)
```

- We decide on 2 clusters (k = 2)
- We fit X to the Kmeans algorithm.

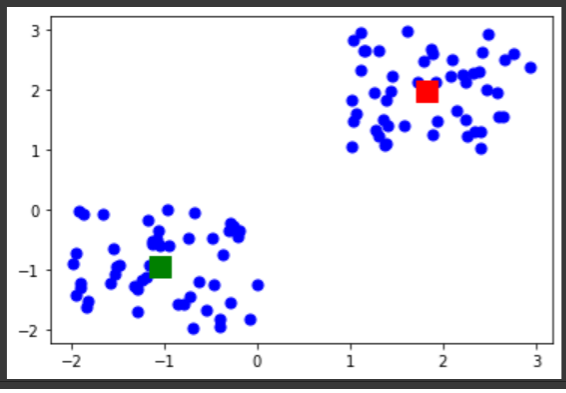
E. CENTROIDS

```
Kmean.cluster_centers_
```

```
array([[ 1.83236473,  1.97474573],
       [-1.03548806, -0.93582751]])
```

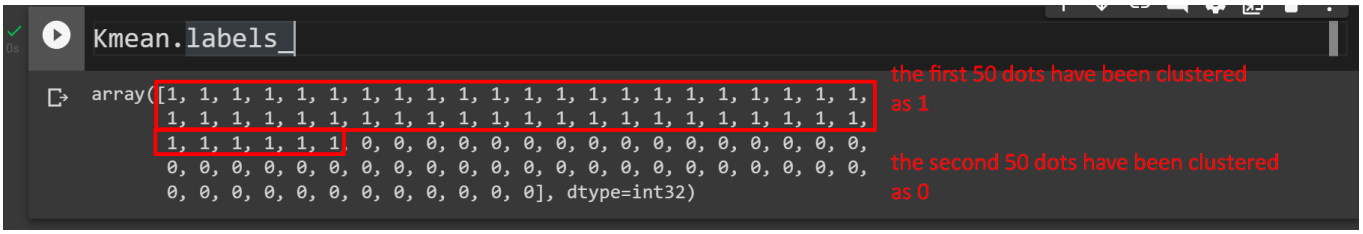
- 2 centroid locations are given:
- [1.83, 1.97] and
- [-1.04, -0.94]

```
[56] plt.scatter(X[ : , 0], X[ : , 1], s =50, c='b')
     plt.scatter(-1.04, -0.94, c='g', s = 200, marker='s')
     plt.scatter(1.83, 1.97, c='r', s = 200, marker='s')
     plt.show()
```

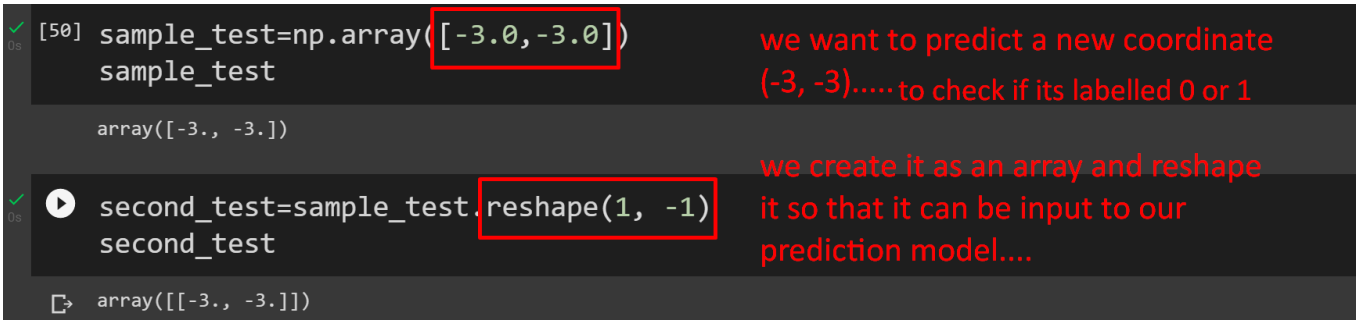


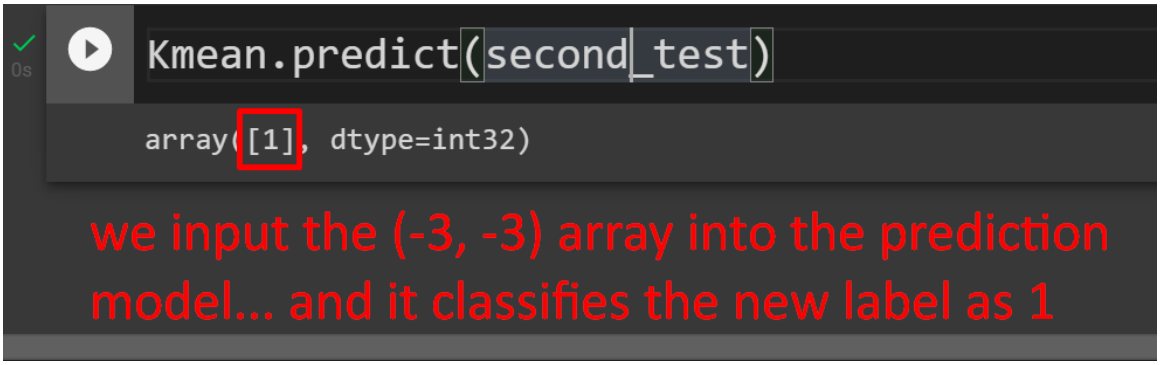
we plot the 2 centroids...

F. LABELING



G. PREDICTING A NEW COORDINATE





ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.