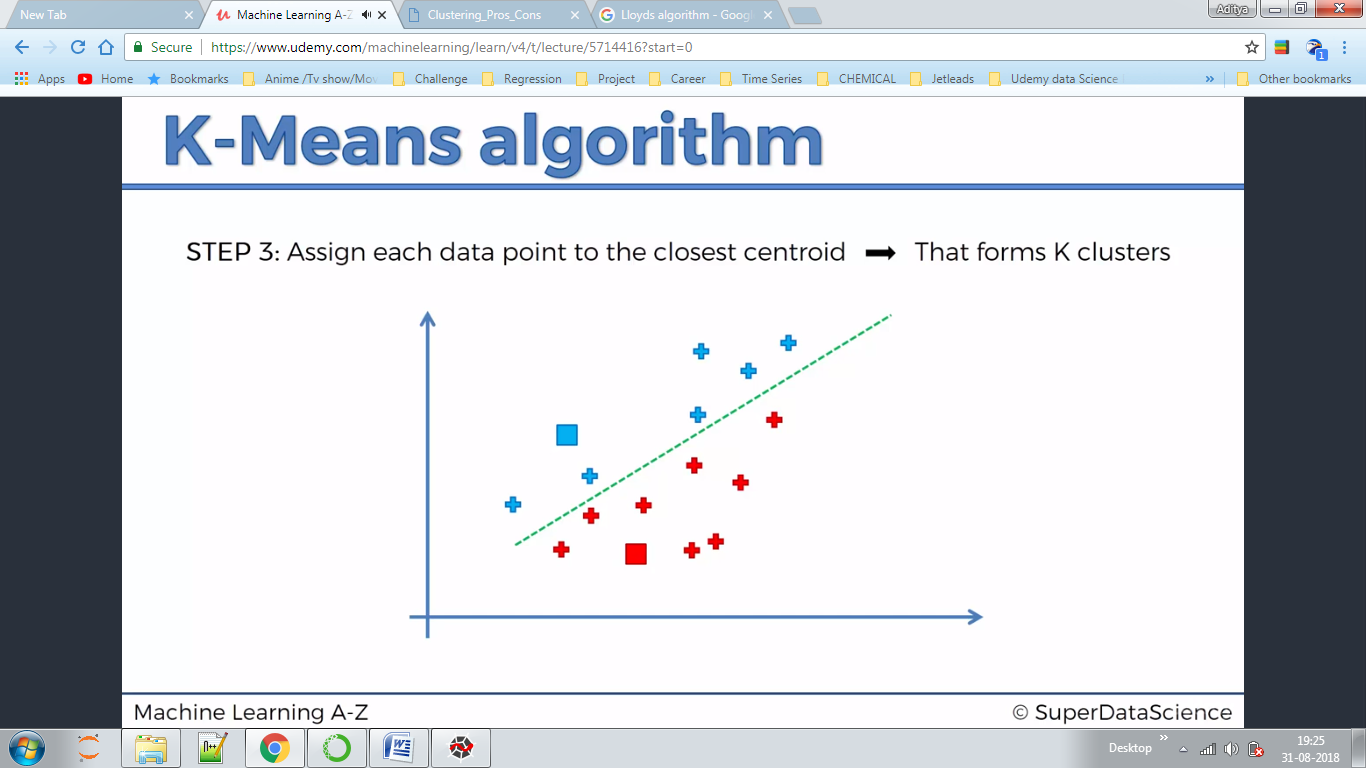


Kmeans clustering

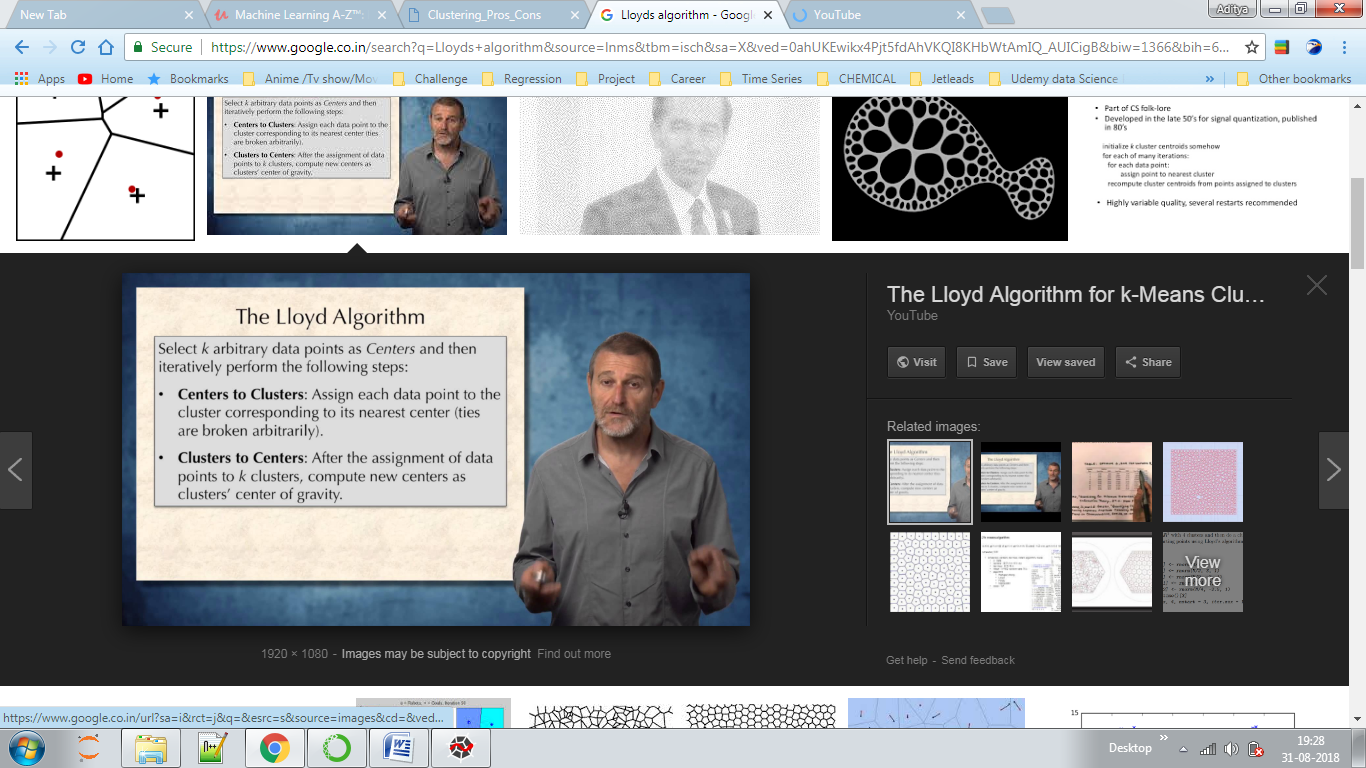
Here we have marked two centroid points and a perpendicular to line joining both of them ,hence we need separate them into different regions.

This model uses Loyds Algortihm

We can decide which distances to use ,default is Euclidean distance.

****

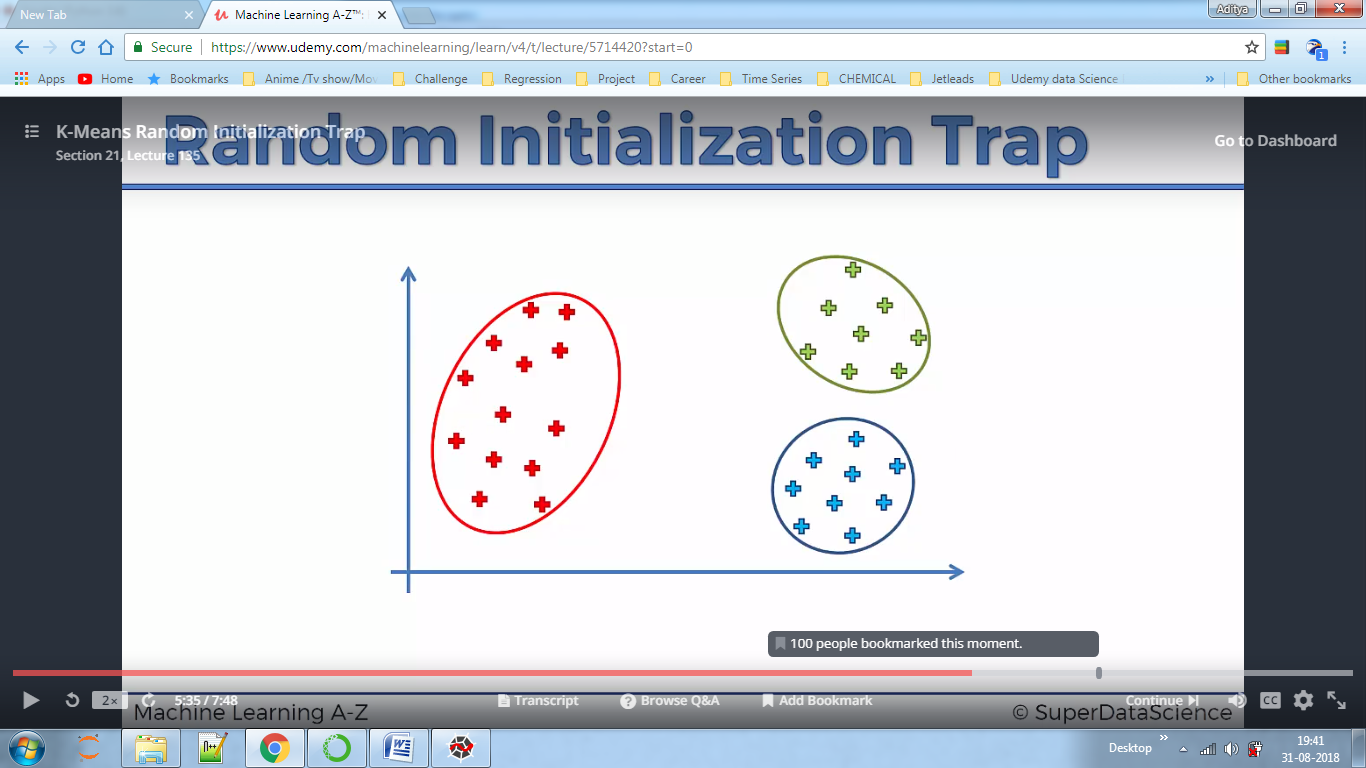
**Lloyds Theorem**

****

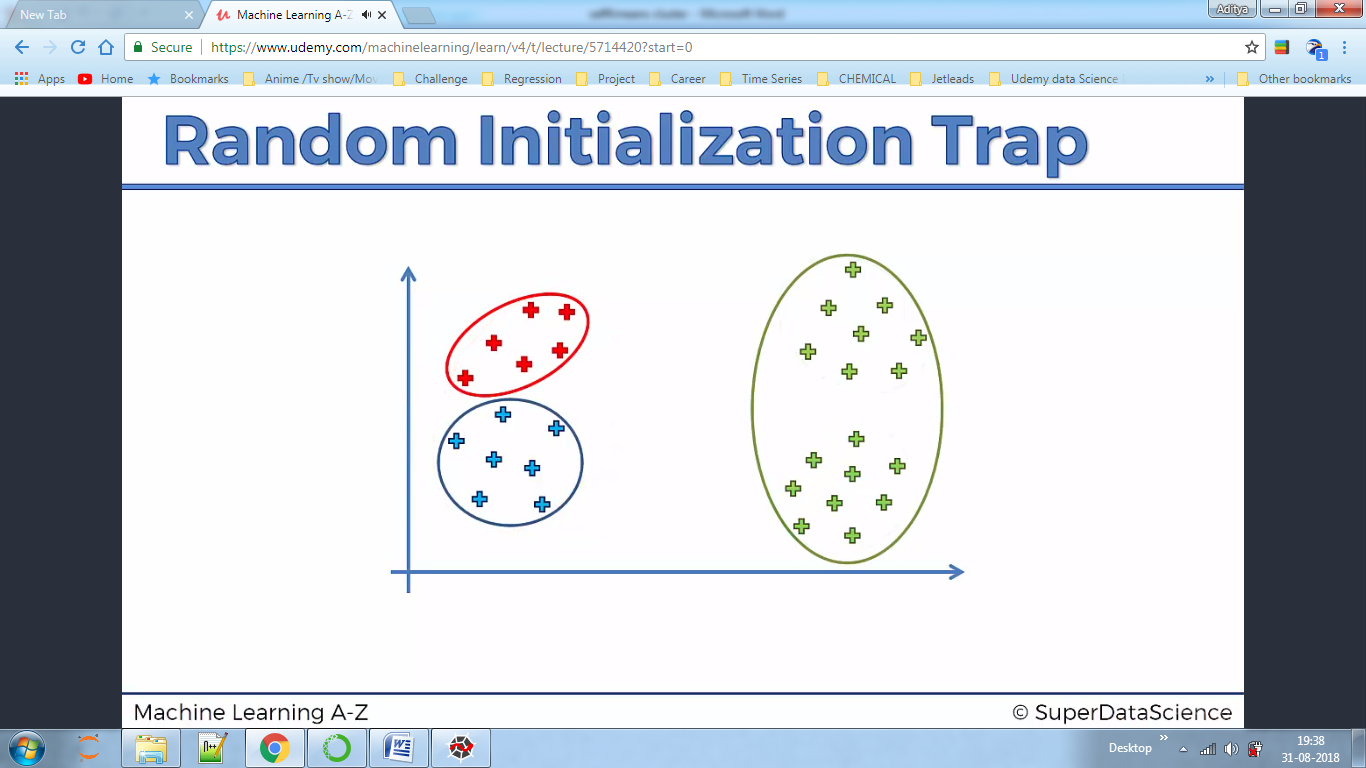
**RANDOM INITIALISATION TRAP**

**-**The starting point, the assumed centroid dictates sometimes the result of the algorithm. It can be corrected using K means ++

E.g. we predicted this:-

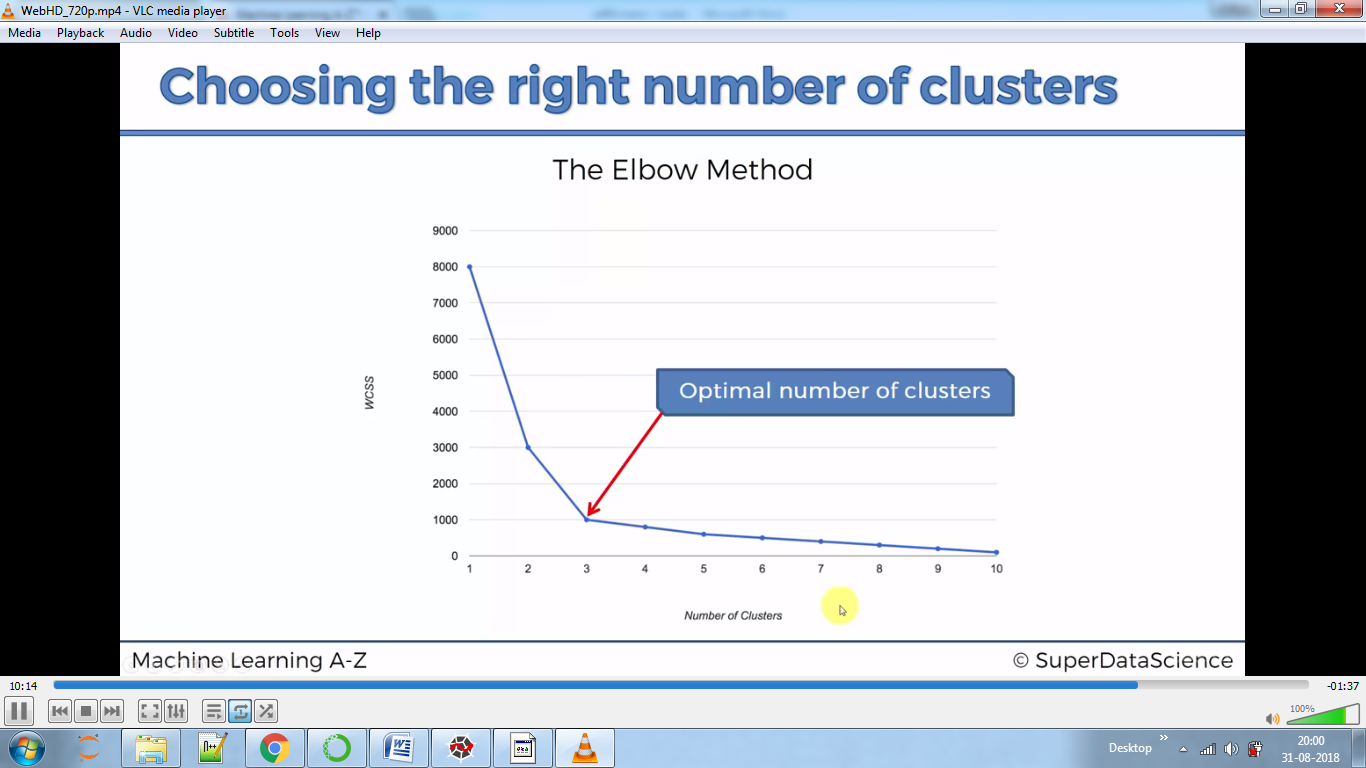


But the answer came out this:-



**Guessing no of Clusters**

This metric is called WCSS-(within clusters sum of squares) metric and is used for guessing clusters in the data .It basically iterates upto say 50 data points and shows us the sum of the Euclidean distances for a data point from centroid(in 2 dimension just like x-axis and y-axis) if we set 1 cluster,2,3...n.

****

**Questions**

Why in the first place are using Kmeans if Kmeans++ is better?

In the code we first predict how many clusters we want ,then predicted values for data points ,then plot them using matplotlib.

plt.scatter(X[y\_kmeans == 0,0], X[y\_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')

In this line we give first argument as an array of all rows X table where y\_kmeans is 0 ,and after comma we give graph xcoordinate as the 0 column of our X which is salary. Here s represents size.