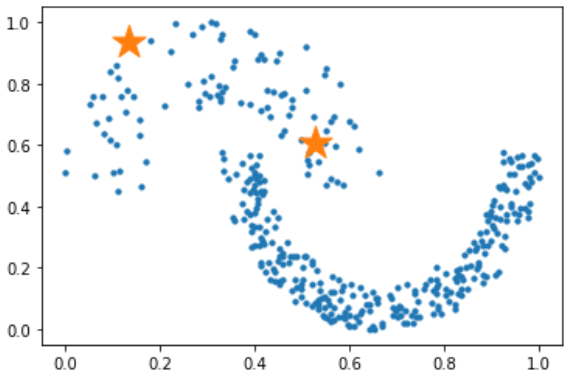
Load samples from "*jain_feats.txt*" into a 2d numpy array **X**. *[For N samples shape should be Nx2]*
Load initial centroids from "*jain_centers.txt*" into another 2d numpy array **centroid_old**. *[For two centroids shape should be 2x2]*
Take another 2d numpy array named **centroid_new** and initialize it with zeros. *[For two centroids shape should be 2x2]*
The initial scatter plot containing **X** and **centroid_old** should look like this:



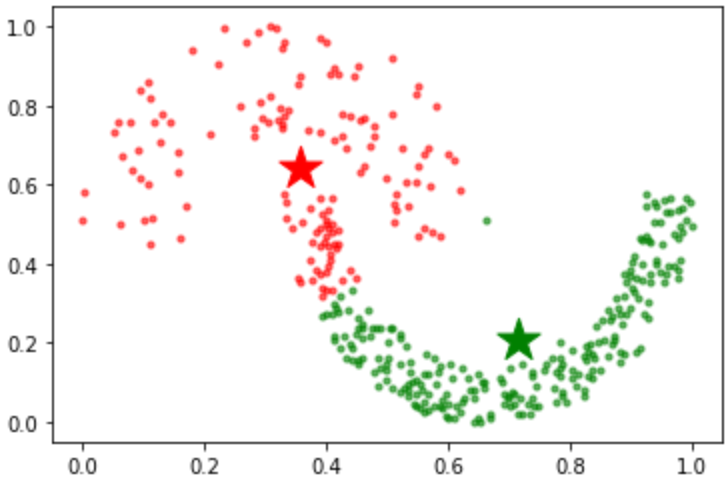
For **e** in iterations(1000):

Assign points to centroids/clusters:
Take a 1D numpy array named **label** with size equals to number of rows in **X**
For each row **i** in **X**:
Take a 1D numpy array named **dist** with size equals to number of rows in **centroid_old**
For each row **j** in **centroid_old**:
Assign **dist**[**j**] := distance between **X**[**i**, :] and **centroid_old**[**j**, :]
label[**i**] := **j**, for which **dist**[**j**] is minimum *[Can easily done by numpy **argmin** method]*

Update Centroids:
For each row **j** in **centroid_new**:
Assign **centroid_new**[**j**] := Average(**X**[**label** == **j**, :]) *[Can easily done by numpy methods]*

Stop condition check:
If:
For each row **j** in **centroid_new**:
Calculate difference between **centroid_new**[**j**] and **centroid_old**[**j**]
If the maximum value among differences found above is less than 1E-7: **STOP**
Else:
centroid_old := **centroid_new**
MOVE to next Iteration

Finally **centroid_old** array holds the final cluster centroids
The final plot should look similar to the following:


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0