**Have a look at the simple k-means program you have been given. Note,it can look for clusters of different ks; but for plotting purposes it fixed to look for k=3. The program can work from a randomly generated set of points or a defined set.**

**Clustering:**

It is the unsupervised face of machine learning given the data as unlabelled. It labels the data points based on how they are groped together.

**K-Means Clustering Algorithm:**

* K- Means clustering is a unsupervised learning algorithm. It is used when the data is not defined in groups or categories i.e. unlabeled data. ***[1]***
* This algorithm is an iterative algorithm that partitions the dataset according to their features into K number of predefined non- overlapping distinct groups. It keeps the data points within the clusters as close as possible and also tries to keep the clusters as far as possible. (Less Intra-cluster distance and More Inter-Cluster distance).
* It clusters ‘n’ data points into ‘k’ clusters where each data points is a part of that cluster with least mean. It runs on Euclidean distance metric calculation in order to find the cluster for the data points. ***[2]***

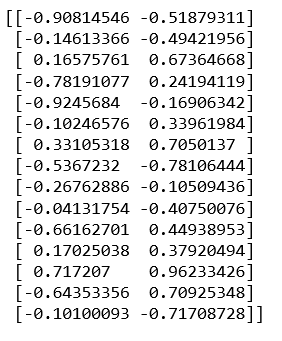
**Q1. Use the init\_board fn to randomly generate 15 points; store this output and set the data variable to it.**

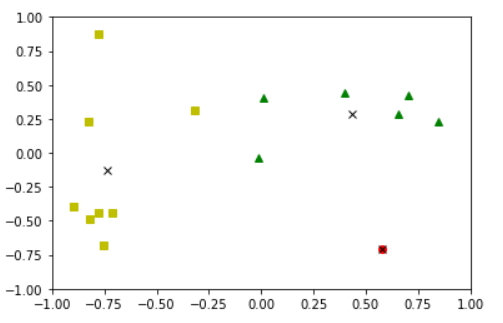
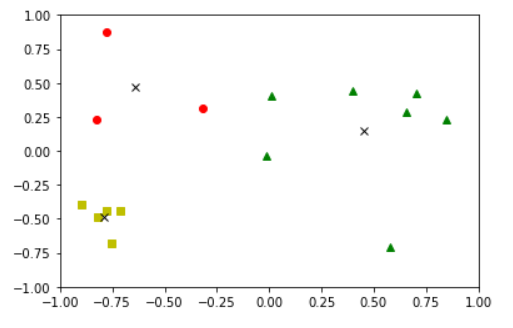
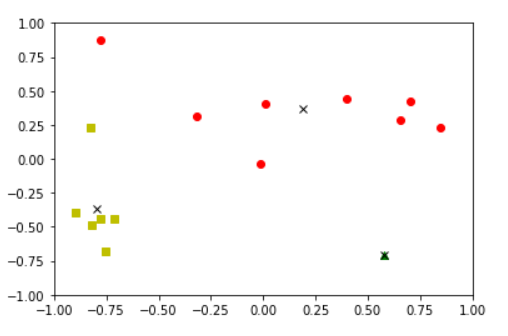
**Now run this set 10 times and note the clusters found by k-means. Report the results of these runs and the extent to which the same clusters are found.**

**Ans1.** Given init\_board function is used to generate 15 different points randomly using the random library.

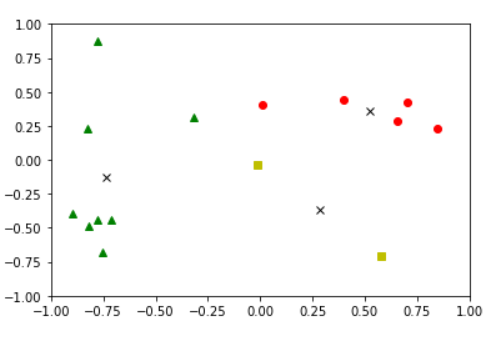
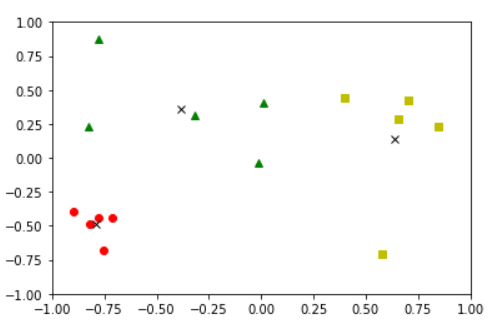
KMeans clustering algorithm is then executed 10 times on the data points and they are clustered into 3 different groups which is specified by parameter K=3. Five different variations of clusters are formed. Two times random data points are clustered in a similar way i.e 5 different variations have 2 exactly similar clusters each.

Random generated data points are shown below:



Category 1 Category 2 Category 3

Category 4 Category 5

This is because the centroids for every cluster are selected randomly and based on the distance from every data point to the centroid, they got clustered. KMeans thus considers noisy data points as well and tries to allocate them into nearby clusters.

=========================================================================================

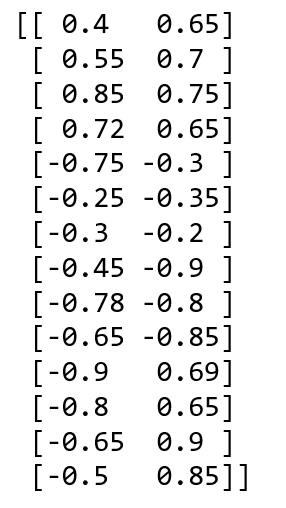
**Q2. Now, create you own set of data, again with 15 points. You should construct this data-set to have very clear clusters (a bit like the simple 6-point example shown).**

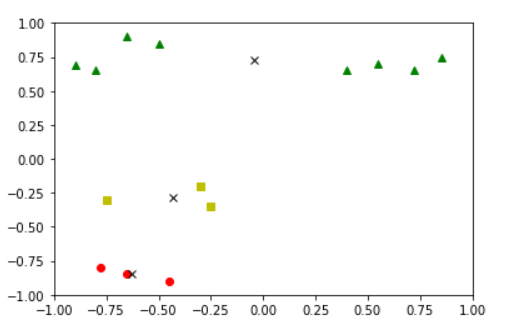
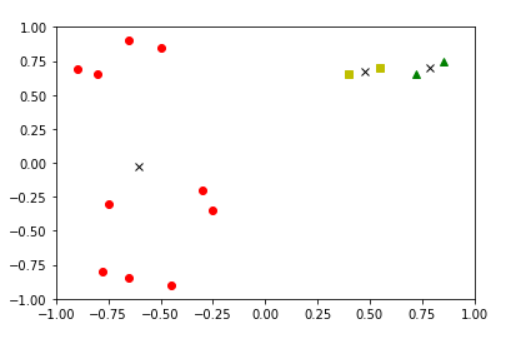
**Now run this set 20 times and note the clusters found by k-means.**

**Report the results of these runs and the extent to which the same clusters are found.**

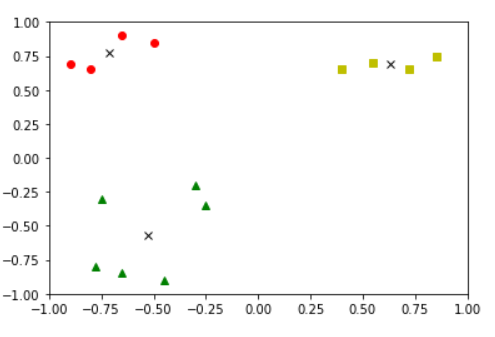
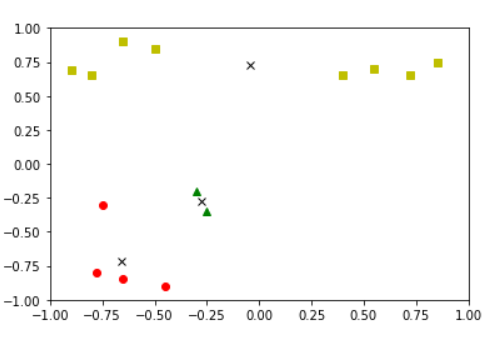
**Ans2.**

* Now, own set of 15 different data points are created. KMeans clustering algorithm is then applied on these 15 different data points to group them among 3 clusters (K=3). The data points are taken carefully such that they gets clustered clearly with more inter-cluster distance ad less intra-cluster distance. That’s why, even the data points are defined beforehand but still they are clustered differently in few cases due to their mean with the centroid points. The various plots shown below depicts clear boundary between the clusters.
* Algorithm is executed 20 number of times and out of 20 runs, 4 different variations of clusters are formed which are attached below.
* Out of 20 iterations, 2 times the clusters formed are of Category 1, 2 times the clusters formed are of Category 2, 13 times they are of Category 3 and 3 times the clusters are of Category 4.
* 4 different variations of the clusters are shown below:
* Set of 15 data points considered are shown below:



Category 1 Category 2

Category 3 Category 4

=========================================================================================

**Q3. Do some research on the problem that k-means produces different answers on different runs.**

**Describe two typical solutions to this problem with references to the literature you read to answer his question.**

**Ans3.** KMeans clustering algorithm produces different answers on different runs because:

* KMeans is a probabilistic technique. It uses random starting values for the centroid.That's why in such a optimization/machine learning problems, multiple iterations are run for the KMeans algorithm and a validation data set can be sued if possible. ***[3]***
* KMeans clustering algorithm places the initial seeds (cluster centers) randomly. So each run of the algorithm will have a different initial set of seed locations, and as such (slightly) different outcomes. The algorithm is often executed multiple times and there is an error measurement calculated as the mean square distance of each point to the cluster centroid to which it belongs. ***[4]***
* Starting with predefined initial seeds instead of randomly selected seeds for the centroids will ensure a consistent answer.
* There is a certain random part in the algorithm that repeatedly uses the random starting points to find the global solution. But due to the pseudo-random nature of the algorithm,slightly different clusters are obtained each time. ***[5]***
* KMeans may not find the global optimum value (random initial points).
* KMeans may not find the global optimum value for number of data points in a cluster.
* As the location of the initial centroids is random, results may not be comparable and show lack of consistency. Also, K-means produces clusters with almost equal number of data points in it, although the data might be different completely and it’s very sensitive to outliers and noisy data.

Two typical solutions to this problems:

1. **KMeans++ Algorithm and Single Pass Seed Selection (SPSS)**: ***[6]***

This helps in robust seed selection to improve the clustering results.

Single Pass Seed Selection, which generates single and optimal solution for the initial seeds and this technique is insensitive to the outliers.

This algorithm selects the start seeds with particular probabilities.

This algorithm selects the high density point as the first centroid and also it calculates the minimum distance that separates the centroid.

The major tasks for Single Pass Seed Selection is to:

* Select optimal centroids
* Generate single clustering solution instead of different results on different runs.

It is a method of initialization to the KMeans algorithm. It first initializes the seed and based on high density, distance is calculated to separate the centroid.

Step 1: Initialize the first centroid with a point which is close to more number of other points in the data set.

Step 2: Calculate the distance matrix(Distance between all data points)

Step 3: Assume that m (total number of points) points are distributed uniformly to k clusters then each cluster is expected to contain m/k points. Compute the sum of the distances from the selected point (in Step1) to first m/k nearest points and assume it as y.

Step4: Find the index of minimum value of Sum from Step3 and find highest density point X.

Step5: Add the high density point to the Centroid set as the first centroid.

Step6: For each point Xi, calculate distance (Xi) to be the distance between Xi and the nearest point in Centroid set.

Step7: Calculate the sum of distances of first m/k nearest points from the X as y.

Step8: Find the unique integer i so that Sum of all squared distances(Xi) >= y > Sum of all distances(Xi i from 1 to i-1).

Step9: Add Xi to Centroid set.

Step10: Repeat steps 5-8 until k centroids are found

* k-means++ algorithm is to careful select the initial seeding for k-means algorithm. However, It has to run multiple number of times for good clustering results but still produces different results in different independent runs.
* The Single Pass Seed Selection yields unique solutions with consistent clustering results using the high density point as the first seed. Thus it is insensitive to the outliers and avoids re runs for the algorithm.

1. **Furthest Point Heuristics (Maxmin): *[7]***

* Initialization done through Maxmin reduces the error rate of clustering on an average from 15% to 6%..
* Repeating it n times reduces the error rate to 1%.
* It selects an arbitrary point as first centroid and then adds points one by one to the centroid set by calculating the maximum distance from the minimum (nearest) already existing centroids.
* For every data object, counter is maintained to its nearest neighbor. When another centroid is chosen, distance from all the data points to the new centroid is calculated and if the distance is less than that of previous, counter is updated. This results in N distance calculations and hence it gets repeated k times, resulting in kN time complexity.
* Further when the new centroid is to be selected, it is done by furthest point from the subset of the data rather than complete dataset.

Algorithm works as follows: ***[8] [9]***

Step1: Choose centroid at random initially.

Step2: Then for every data points, choose next centroid which is farthest from the previously selected centroid.

It fixes the Gaussian problem but it is sensitive to outliers.

* Only the initial point is selected randomly, the rest of the points are selected deterministically. Then for each point in Xi = [Xi,1,Xi,2……..,Xi,m]. Every point is described by m features. A function f(xi,j) is used to calculate the frequency count for every attribute value.
* Then a scoring function is defined (Summation of all function f(xi,j)) to evaluate the data point and point with highest score will be chosen as first point and remaining are selected the same way.
* The points that contain attribute values with higher frequencies most probably becomes the centers.All the rest of the points are selected deterministically. This shows that fixed clustering results could be achieved from our new initialization procedure based k-modes clustering algorithm.
* The time complexity to choose first point is : O(n) with two scans over the dataset:

1. m number of hash tables are constructed to store attribute values and their frequencies.

2. Frequency count of an attribute value from the corresponding hash table is fetched in O(1).

Hence the time complexity for data point with largest score is O(n).

Thus, the overall time complexity of new farthest-point heuristic is still O(nk).

=========================================================================================

**REFERENCES**

1. EDUCBA, "K- Means Clustering Algorithm", 2019 [Online], Available at: <https://www.educba.com/k-means-clustering-algorithm/> [Accessed Nov. 2, 2019]
2. IntelliPaat, "What Is K means clustering Algorithm in Python", 2019 [Online], Available at: <https://intellipaat.com/blog/k-means-clustering/> [Accessed Nov. 2, 2019]
3. J.D’Errico, “MathWorks”, 2019 [Online], Available at: <https://uk.mathworks.com/matlabcentral/answers/167134-why-kmeans-gives-different-results-each-time> [Accessed Nov. 2, 2019]
4. Harthur, “different runs of k-means clustering result in different outputs”, 2019 [Online], Available at: <https://github.com/harthur/clustering/issues/9> [Accessed Nov. 2, 2019]
5. Ansari, “Matlab: Kmeans gives different results each time”, 2019 [Online], Available at: <https://stackoverflow.com/questions/12137282/matlab-kmeans-gives-different-results-each-time> [Accessed Nov. 2, 2019]
6. International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 5, Oct 2011, “Robust seed selection algorithm for k-means type algorithms ”, 2011 , Available at: <https://arxiv.org/ftp/arxiv/papers/1202/1202.1585.pdf> [Accessed Nov. 2, 2019]
7. ScienceDirect: Pattern Recognition: Volume 93, September 2019, Pages 95-112, “How much can k-means be improved by using better initialization and repeats?”, 2019 , Available at: <https://www.sciencedirect.com/science/article/pii/S0031320319301608> [Accessed Nov. 2, 2019]
8. Machine Learning Department, School of Computer Science, Carnegie Mellon University, “Clustering (K-Means)”, 2019 , Available at: <https://www.cs.cmu.edu/~mgormley/courses/10601-s17/slides/lecture15-cluster.pdf> [Accessed Nov. 2, 2019]
9. Zengyou He, “Farthest-Point Heuristic based Initialization Method for K-Modes Clustering ”, Available at: <https://pdfs.semanticscholar.org/ab3c/05b5e74399c38501291703e373ded224a464.pdf> [Accessed Nov. 2, 2019]