**Machine Learning Solution Worksheet 2**

**12. Is K sensitive to outliers?**

**Ans.** The k-means algorithm updates the cluster centers by taking the average of all the data points that are closer to each cluster center. When all the points are packed nicely together, the average makes sense. However, when you have outliers, this can affect the average calculation of the whole cluster. As a result, this will push your cluster center closer to the outlier.

An example, is the average of the salaries of the following people:

$50k, $20k, $35k, $65k and $1 Million

The average ends up being ($50k + $20k + $35k + $65k + $1MM) / 5 = $1170k / 5 = $234k.

If we did not have the $1MM outlier, the average would have been ($50k + $20k + $35k + $65k) / 4 = $170k / 4 = $42.5k.

Note that the two average results are wildly different from one another.

Thus we can say k-means can be quite sensitive to outliers in your data set. The reason is simply that k-means tries to optimize the **sum of squares**. And thus a large deviation (such as of an outlier) gets a lot of weight.

If there is a noisy data set with outliers, you might be better off using an algorithm that has specialized noise handling such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Note the "N" in the acronym: Noise. In contrast to e.g. k-means, but also many other clustering algorithms, DBSCAN can decide to not cluster objects that are in regions of low density.

**13. Why is K means better?**

**Ans**. K-Means clustering is a very popular and simple clustering technique. The main objective of K-Means clustering is to group the similar data points into clusters. Here, ‘K’ means the number of clusters, which is predefined.   
Let’s take some example,  
We have a dataset which has three features (three variables) and a total of 200 observations. Let’s assume K= 3 (number of clusters). As we know, clustering will group similar data points into one group. Hence, our dataset has been divided into three clusters with each cluster having similar points.  
From the above figure, our dataset has divided into three clusters such as cluster 1, cluster 2 and cluster 3.  
OK, how does the K-Means algorithm know the similar point in order to form the clusters?  
For every cluster, it assigns a random point called centroid which is called the central point of clusters. From the below figure, we can see the centroids for each cluster.  
K-Means clustering is also called centroid based clustering. If you say K =5, then we can get five centroids and say K = 4, then we have four centroids.  
In our above example, a number of clusters are 3 (K), so it has three centroids. If we find the centroids, then the nearest data points would be assigned to each centroid. In this case, three centroids such as C1, C2, and C3 were created and all the nearest data points would be assigned to each centroid and form the cluster. From the above picture, if we add all the data points in all three clusters would give our total observations of 200 data points. Please note no data points belong to two clusters.   
Here, the important task is to minimize the sum of distances of the points to their respective centroid.  
In simpler terms, the goal of the K-Means cluster is to find out the centroid for each cluster and get the data points to be assigned to the nearest centroid.   
The K-means clustering algorithm is used to group unlabeled data set instances into clusters based on similar attributes. It has a number of advantages over other types of machine learning models, including the linear models, such as logistic regression and [Naive Bayes](https://automaticaddison.com/naive-bayes-algorithm-from-scratch-machine-learning/).  
  
**Here are the advantages:**  
  
**Unlabeled Data Sets**A lot of real-world data comes unlabeled, without any particular class. The benefit of using an algorithm like K-means clustering is that we often do not know how instances in a data set should be grouped.   
For example, consider the problem of trying to group viewers of Netflix into clusters based on similar viewing behavior. We know that there are clusters, but we do not know what those clusters are. Linear models will not help us at all with these sorts of issues.

**Nonlinearly Separable Data**consider the data set below containing a set of three concentric circles. It is nonlinearly separable. In other words, there is no straight line or plane that we could draw on the graph below that can easily discriminate the colored classes red, blue, and green. Using K-means clustering and converting the coordinate system below from Cartesian coordinates to Polar coordinates, we could use the information about the radius to create concentric clusters.

**Simplicity**The meat of the K-means clustering algorithm is just two steps, the **cluster assignment** step and the **move centroid** step. If we’re looking for an unsupervised learning algorithm that is easy to implement and can handle large data sets, K-means clustering is a good starting point.

**Availability**Most of the popular machine learning packages contains an implementation of K-means clustering.

**Speed**based on my experience using K-means clustering, the algorithm does its work quickly, even for really big data sets.

#### High Performance

K-Means algorithm has linear time complexity and it can be used with large datasets conveniently. With unlabeled big data K-Means offers many insights and benefits as an unsupervised clustering algorithm.

#### Easy to Use

K-Means is also easy to use. It can be initialized using default parameters in the Scikit-Learn implementation. According to this approach, parameters like number of clusters (8 by default), maximum iterations (300 by default), initial centroid initialization (10 by default) can easily be adjusted later on to suit the task goals.

#### Unlabeled Data

This one is a general unsupervised machine learning algorithm that also applies to K-Means.  
  
If your data has no labels (class values or targets) or even column headers, K-Means will still successfully cluster your data. This is an example to machine learning harvesting data and extracting useful insights from data that can be totally useless to human eye.  
  
Customer segmentation, scientific categorization, logistic optimization (identifying inventories or optimizing routes), user suggestions, patient management, trial management and fraud detection are just a few example use cases.

#### Result Interpretation

K-Means returns clusters which can be easily interpreted and even visualized. This simplicity makes it highly useful in some cases when you need a quick overview of the data segments.  
  
Additionally, inertia values produced by K-Means algorithm can be meaningful to interpret as well. K-Means inertia sum of squared means for each point to their respective cluster center (centroid). Higher inertia values can be helpful to question cluster number or algorithm's inner workings such as initialization or maximum iteration.  
You can read more details about K-Means settings in the following link:  
[Optimization of K-Means parameters](https://holypython.com/k-means/k-means-optimization-parameters/).

**Q14. Is K means a deterministic algorithm?**

 Ans. A **deterministic algorithm** is an algorithm that, given a particular input, will always produce the same output, with the underlying machine always passing through the same sequence of states. Deterministic algorithms can be run on real machines efficiently.

A deterministic algorithm computes a mathematical function; a function has a unique value for any input in its domain, and the algorithm is a process that produces this particular value as output.

 K-Means takes data points as input and groups them into k clusters. This process of grouping is the training phase of the learning algorithm. **The result would be a model that takes a data sample as input and returns the cluster that the new data point belongs to, according the training that the model went through.**

The basic k-means clustering is based on a non-deterministic algorithm. This means that running the algorithm several times on the same data, could give different results.

The non-deterministic nature of K-Means is due to its random selection of data points as initial centroids. The key idea of the algorithm is to select data points which belong to dense regions and which are adequately separated in feature space as the initial centroids.