CLUSTERING QUESTIONS

Can decision trees be used for performing clustering?

Decision trees can also be used to for clusters in the data, but clustering often generates natural clusters and is not dependent on any objective function.

What is Objective Function?

- The function we want to minimize or maximize is called the objective function, or criterion. When we are minimizing it, we may also call it the cost function, loss function, or error function - these terms are synonymous.
- The cost function is used more in optimization problem and loss function is used in parameter estimation.
- The loss function (or error) is for a single training example, while the cost function is over the entire training set (or mini-batch for mini-batch gradient descent).

Therefore, a loss function is a part of a cost function which is a type of an objective function. Objective function, cost function, loss function: are they the same thing?

1. **Loss function** is usually a function defined on a data point, prediction and label, and measures the penalty. For example: square loss $l(f(x_i|\theta),y_i)=(f(x_i|\theta)-y_i)2$, used in linear Regression
2. **Cost function** is usually more general. It might be a sum of loss functions over your training set plus some model complexity penalty (regularization). For example: Mean Squared Error $MSE(\theta)=1N\sum Ni=1(f(x_i|\theta)-y_i)2$
3. **Objective function** is the most general term for any function that you optimize during training. For example, a probability of generating training set in maximum likelihood approach is a well-defined objective function, but it is not a loss function nor cost function (however you could define an equivalent cost function). For example: MLE is a type of objective function (which you maximize)
4. **Error function -** Backpropagation; or automatic differentiation, is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. This technique is also sometimes called backward propagation of errors, because the error is calculated at the output and distributed back through the network layers.

Which of the following is the most appropriate strategy for data cleaning before performing clustering analysis, given less than desirable number of data points:

Removal of outliers is not recommended if the data points are few in number. In this scenario, capping and flouring of variables is the most appropriate strategy.

What is the minimum no. of variables/ features required to perform clustering?

At least a single variable is required to perform clustering analysis. Clustering analysis with a single variable can be visualized with the help of a histogram.

For two runs of K-Mean clustering is it expected to get same clustering results?

K-Means clustering algorithm instead converses on local minima which might also correspond to the global minima in some cases but not always. Therefore, it's advised to run the K-Means algorithm multiple times before drawing inferences about the clusters.

However, note that it's possible to receive same clustering results from K-means by setting the same seed value for each run. But that is done by simply making the algorithm choose the set of same random no. for each run.

Is it possible that Assignment of observations to clusters does not change between successive iterations in K-Means

When the K-Means algorithm has reached the local or global minima, it will not alter the assignment of data points to clusters for two successive iterations.

Which of the following can act as possible termination conditions in K-Means?

1. For a fixed number of iterations.
2. Assignment of observations to clusters does not change between iterations. Except for cases with a bad local minimum.
3. Centroids do not change between successive iterations.
4. Terminate when RSS falls below a threshold.

What is Residual Sum of Square?

- The residual sum of squares or RSS is the square distance of each vector from its centroid summed over all points
- RSS is objective function of the k-means clustering minimization
- Since the number the points N is fixed, RSS is equivalent to minimizing the average square distance, a measure of how well the centroids represent their points in their clusters
- First, RSS decreases in the reassignment step: each point p is assigned to its closest centroid, so the distance it contributes to RSS decreases
- Second, it decreases in the recomputation step because the new centroid is the minimum of the RSSr where point p was reassigned to cluster Cr

Time complexity of the k-means clustering algorithm

- O(N) a linear time algorithm
- Most time is computing distances between a point and a centroid, such a computation takes O(1)
- The reassignment of a point to one of the k centroids takes constant time as k is a constant
- Overall we can compute kN pairwise distances
- If we perform L iterations (one iteration is reassignment of all the points) then the overall time is O(LkN) which is O(N) as Li and k are constants

Which of the following clustering algorithms suffers from the problem of convergence at local optima?

K-Means clustering algorithm and EM clustering algorithm has the drawback of converging at local minima.

Which of the following algorithm is most sensitive to outliers?

K-Means clustering algorithm is most sensitive to outliers as it uses the mean of cluster data points to find the cluster center.

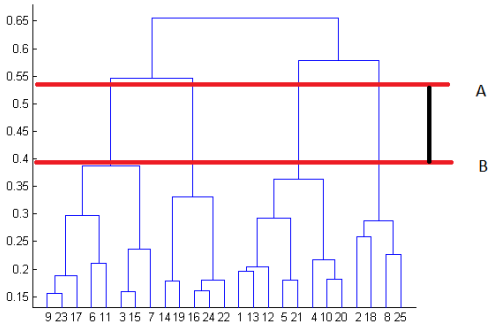
How can Clustering (Unsupervised Learning) be used to improve the accuracy of Linear Regression model (Supervised Learning):

1. Creating different models for different cluster groups.
2. Creating an input feature for cluster ids as an ordinal variable.
3. Creating an input feature for cluster centroids as a continuous variable.
4. Creating an input feature for cluster size as a continuous variable.

What could be the possible reason(s) for producing two different dendrograms using agglomerative clustering algorithm for the same dataset?

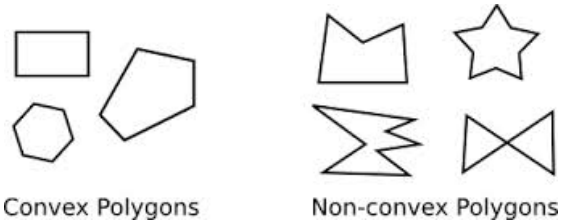
1. Proximity function used
2. of data points used
3. of variables used

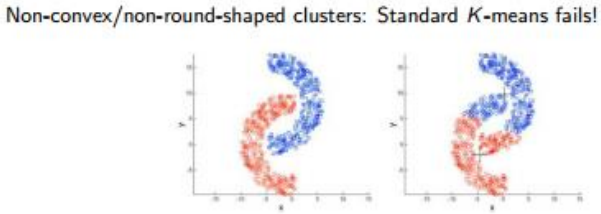
What is the most appropriate no. of clusters for the data points represented by the following dendrogram: 4

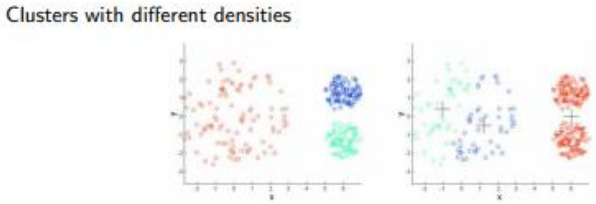


In which of the following cases will K-Means clustering fail to give good results?

1. Data points with outliers **(will give BAD result)**
2. Data points with different densities **(will give BAD result)**
3. Data points with round shapes **(will give GOOD result)**
4. Data points with non-convex shapes **(will give BAD result)**





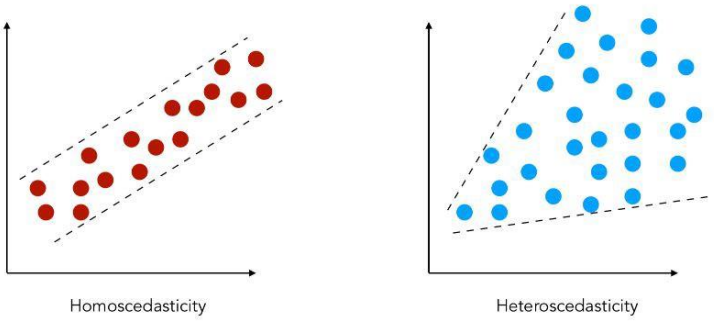


Finding dissimilarity between two clusters in hierarchical clustering?

1. Single linkage
2. Complete linkage
3. Average linkage

How Clustering is affected by Multicollinearity and Heteroscedasticity?

Clustering analysis is not negatively affected by heteroscedasticity, but the results are negatively impacted by multicollinearity of features/ variables used in clustering as the correlated feature/ variable will carry extra weight on the distance calculation than desired.



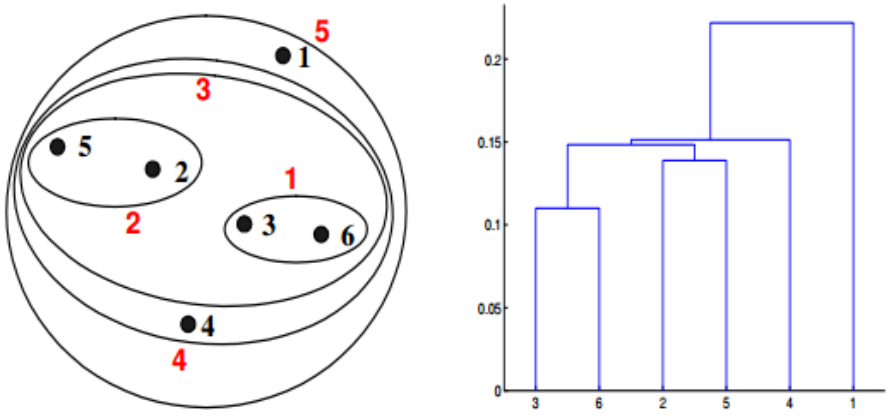
Given, six points with the following attributes, following clustering representations and dendrogram depicts the use of MIN or Single link proximity function in hierarchical clustering:

point	x coordinate	y coordinate
p1	0.4005	0.5306
p2	0.2148	0.3854
p3	0.3457	0.3156
p4	0.2652	0.1875
p5	0.0789	0.4139
p6	0.4548	0.3022

Table : X-Y coordinates of six points.

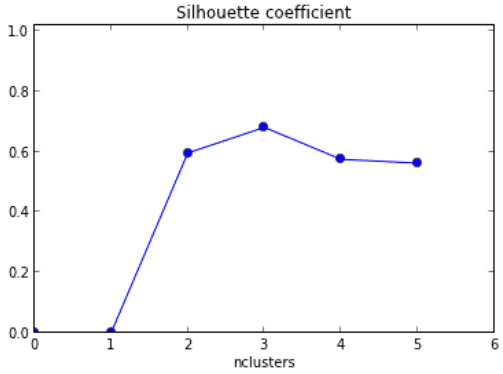
	p1	p2	p3	p4	p5	p6
p1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
p3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
p4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

Table : Distance Matrix for Six Points



For the single link or MIN version of hierarchical clustering, the proximity of two clusters is defined to be the minimum of the distance between any two points in the different clusters. For instance, from the table, we see that the distance between points 3 and 6 is 0.11, and that is the height at which they are joined into one cluster in the dendrogram. As another example, the distance between clusters {3, 6} and {2, 5} is given by dist({3, 6}, {2, 5}) = min(dist(3, 2), dist(6, 2), dist(3, 5), dist(6, 5)) = min(0.1483, 0.2540, 0.2843, 0.3921) = 0.1483.

What should be the best choice of no. of clusters based on the following results: 3



The silhouette coefficient is a measure of how similar an object is to its own cluster compared to other clusters. Number of clusters for which silhouette coefficient is highest represents the best choice of the number of clusters.

Which of the following is/are valid iterative strategy for treating missing values before clustering analysis?

1. Imputation with mean
2. Nearest Neighbor assignment
3. Imputation with Expectation Maximization algorithm (It is iterative in its functioning)

Assume, you want to cluster 7 observations into 3 clusters using K-Means clustering algorithm. After first iteration clusters, C1, C2, C3 has following observations:

C1: {(2,2), (4,4), (6,6)}

C2: {(0,4), (4,0)}

C3: {(5,5), (9,9)}

What will be the cluster centroids if you want to proceed for second iteration?

1. Finding centroid for data points in cluster C1 = ((2+4+6)/3, (2+4+6)/3) = (4, 4)
2. Finding centroid for data points in cluster C2 = ((0+4)/2, (4+0)/2) = (2, 2)
3. Finding centroid for data points in cluster C3 = ((5+9)/2, (5+9)/2) = (7, 7)
   Hence, C1: (4,4), C2: (2,2), C3: (7,7)

Assume, you want to cluster 7 observations into 3 clusters using K-Means clustering algorithm. After first iteration clusters, C1, C2, C3 has following observations:

C1: {(2,2), (4,4), (6,6)}

C2: {(0,4), (4,0)}

C3: {(5,5), (9,9)}

What will be the Manhattan distance for observation (9, 9) from cluster centroid C1. In second iteration.

Manhattan distance between centroid C1 i.e. (4, 4) and (9, 9) = (9-4) + (9-4) = 10

If two variables V1 and V2, are used for clustering. Which of the following are true for K means clustering with k =3?

1. **If V1 and V2 has a correlation of 1, the cluster centroids will be in a straight line**
2. **If V1 and V2 has a correlation of 0, the cluster centroids will be in straight line**

If the correlation between the variables V1 and V2 is 1, then all the data points will be in a straight line. Hence, all the three cluster centroids will form a straight line as well.

Feature scaling is an important step before applying K-Mean algorithm. What is reason behind this?

Feature scaling ensures that all the features get same weight in the clustering analysis. Consider a scenario of clustering people based on their weights (in KG) with range 55-110 and height (in inches) with range 5.6 to 6.4. In this case, the clusters produced without scaling can be very misleading as the range of weight is much higher than that of height. Therefore, its necessary to bring them to same scale so that they have equal weightage on the clustering result.

Disadvantages about K-Mean Clustering?

1. K-means is extremely sensitive to cluster center initializations
2. Bad initialization can lead to Poor convergence speed
3. Bad initialization can lead to bad overall clustering

Following can be applied to get good results for K-means algorithm corresponding to global minima?

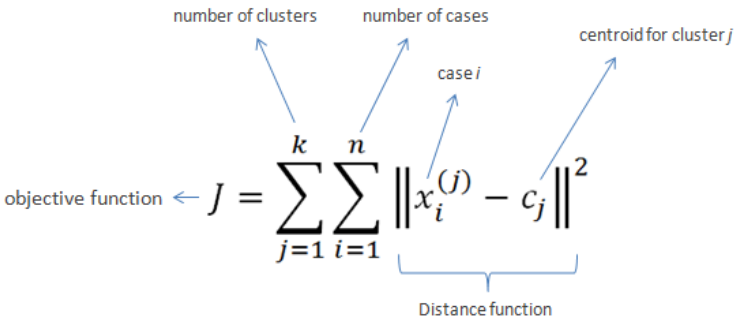
1. Try to run algorithm for different centroid initialization
2. Adjust number of iterations
3. Find out the optimal number of clusters

If you are using Multinomial mixture models with the expectation-maximization algorithm for clustering a set of data points into two clusters, which of the assumptions are important:

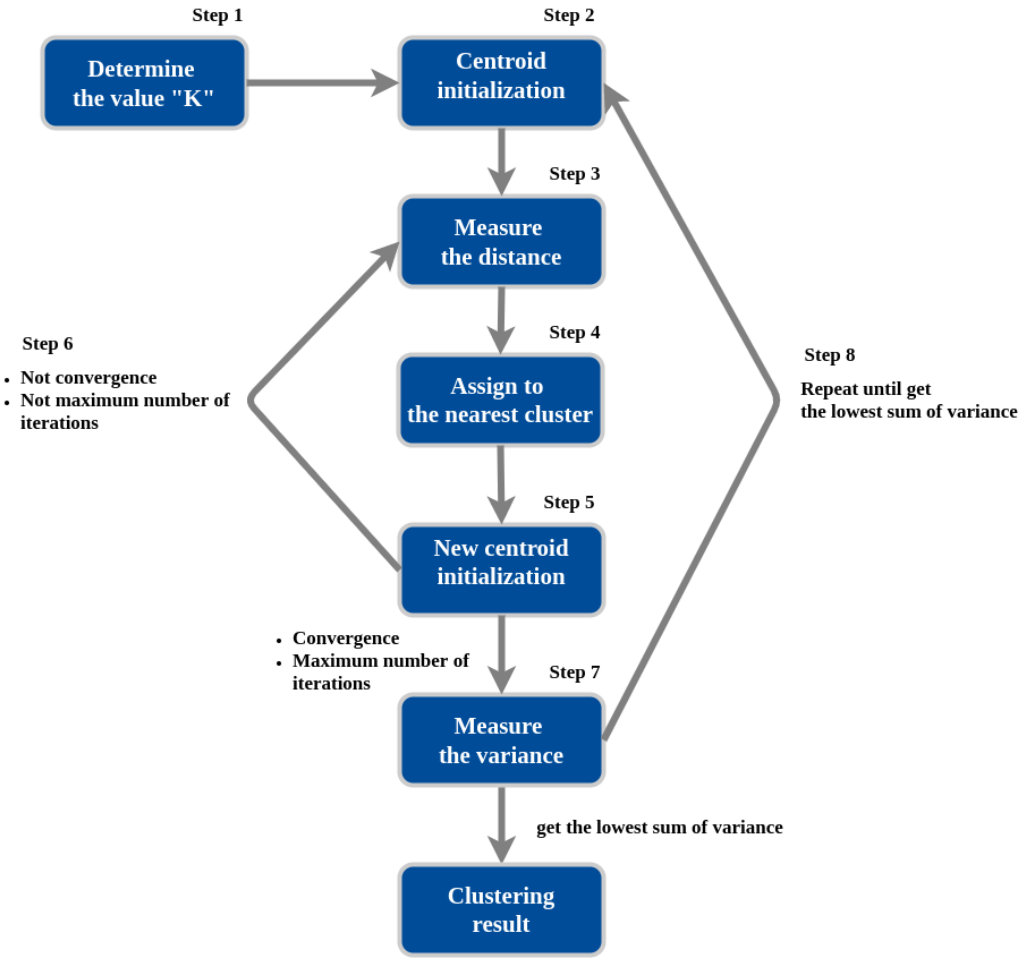
In EM algorithm for clustering its essential to choose the same no. of clusters to classify the data points into as the no. of different distributions they are expected to be generated from and also the distributions must be of the same type.

Explain the steps of k-Means Clustering Algorithm

- **K-Means** clustering intends to partition n objects into k clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of the greatest possible distinction.
- The best number of clusters *k* leading to the greatest separation (distance) is not known as a priori and must be computed from the data.
- The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:



1. Clusters the data into *k* groups where *k* is predefined.
2. Select *k* points at *random* as cluster centers.
3. Assign objects to their closest cluster center according to the *Euclidean distance* function.
4. Calculate the *centroid* or *mean* of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.



Explain what is *k-Means Clustering*?

- **k-means clustering** is a method of *vector quantization* that aims to partition n observations into k clusters in which each observation belongs to the *cluster* with the nearest *mean*.
- k-means clustering minimizes **within-cluster variances**.
- **Within-cluster-variance** is simple to understand **measure of compactness**. So basically, the objective is to find the most *compact* partitioning of the data set into k partitions.

Algorithm	K-Means
Type	Unsuperised Machine Learning
Use	To find groups of rows(clusters) in data
Cost Function	Distortion within clusters(distances of each point from the cluster center)
Goodness of Fit	Within cluster SSE (km.inertia_)
Hyperparameters	n_init, max_iter, init
Function in R	km=**kmeans**(x=ClusterData, centers=3, iter.max = 10) Km
Function in Python	from sklearn.cluster import **KMeans** km = **KMeans**(n_clusters=2,n_init=10, max_iter=300, random_state=0) km.fit(X)

What are some Stopping Criteria for k-Means Clustering?

1. Convergence. No further changes, points stay in the same cluster.
2. The maximum number of iterations. When the maximum number of iterations has been reached, the algorithm will be stopped. This is done to limit the runtime of the algorithm.
3. Variance did not improve by at least x
4. Variance did not improve by at least x * initial variance

Explain some cases where k-Means clustering fails to give good results

1. k-means has trouble clustering data where clusters are of various sizes and densities.
2. Outliers will cause the centroids to be dragged, or the outliers might get their own cluster instead of being ignored. Outliers should be clipped or removed before clustering.
3. If the number of dimensions increase, a distance-based similarity measure converges to a constant value between any given examples. Dimensions should be reduced before clustering them.

How is Entropy used as a Clustering Validation Measure?

Entropy is a measure of the purity of the cluster with respect to the given class label. Thus, if each cluster consists of objects with a single class label, the entropy value is 0. As the objects in a cluster become more diverse, the entropy value increases.

Using entropy measure to validate the class labels tends to favor k-means which produce clusters in relatively uniform size. This effect is more significant in the situation that the data have highly imbalanced true clusters.

So, using entropy measure for validating k-means clustering can lead to the results being misleading.

How would you Pre-Process the data for k-Means?

Some pre-processing steps to follow are:

1. If the variables are of incomparable units, then the variables should be standardized.
2. Even if the variables are of the same units but show quite different variances then it is a good idea to standardize them. Since k-means clustering produces more or less round clusters, it puts more weight on variables with smaller variance, so the clusters will tend to be separated along with variables with greater variance.
3. k-means clustering results are sensitive to the order of objects in the dataset, so it is good to randomize the dataset and try clustering many different times.

Algorithm	**Hierarchical Clustering**
Type	Unsuperised Machine Learning
Use	To find groups of rows(clusters) in data
Cost Function	Linkage method (ward, average, complete)
Goodness of Fit	Dendrogram
Hyperparameters	n_clusters=2, affinity = 'euclidean', linkage = 'ward'

Function in R	# Computing the distance matrix
	DistanceMatrix=**dist**(ClusterData)
	
	# Creating the Hirarchical clusters
	Hcluster=**hclust**(DistanceMatrix)
	
	# Cutting the dendrogram to get 3 clusters using parameter k
	ClusterID=**cutree**(Hcluster, k=3)

Function in Python	from sklearn.cluster import **AgglomerativeClustering** hc = **AgglomerativeClustering**(n_clusters=2, affinity = 'euclidean', linkage = 'ward')

Algorithm	DBSCAN
Type	Unsuperised Machine Learning
Use	To find groups of rows(clusters) in data
Cost Function	Distance from core points (eps)
Goodness of Fit	Within Cluster SSE
Hyperparameters	eps = 0.5, minPts = 5

Function in R	# Creating Clusters using DBSCAN algorithm
	library(**dbscan**)
	DBSCANclusters <- **dbscan**(ClusterData, eps = 0.5, minPts = 5)
	print(DBSCANclusters)

Function in Python	from sklearn.cluster import **DBSCAN** db = **DBSCAN**(eps=1, min_samples=5)

IS K-MEANS CLUSTERING SUITABLE FOR ALL SHAPES AND SIZES OF CLUSTERS?

- K-means is not suitable for all shapes, sizes, and densities of clusters. If the natural clusters of a dataset are vastly different from a spherical shape, then K-means will face great difficulties in detecting it.
- K-means will also fail if the sizes and densities of the clusters are different by a large margin. This is mostly due to using SSE as the objective function, which is more suited for spherical shapes.
- SSE is not suited for clusters with non-spherical shapes, varied cluster sizes, and densities.

WHAT ARE THE ISSUES WITH RANDOM INITIALIZATION OF CENTROIDS IN K-MEANS ALGORITHM AND HOW TO OVERCOME IT?

- Initiation of the centroids in a cluster is one of the most important steps of the K-means algorithm. Many times, random selection of initial centroid does not lead to an optimal solution.
- In order to overcome this problem, the algorithm is run multiple times with different random initialisations.
- The sum of squared errors (SSE) are calculated for different initial centroids. The set of centroids with the minimum SSE is selected.
- Even though this is a very simple method, it is not foolproof.
- The results of multiple random cluster initialisations will depend on the dataset and the number of clusters selected, however, that still will not give an optimum output every time.

What are the K-means assumptions

- Variables are all continuous
- Variables have a symmetric distribution (i.e., not skewed)
- Variables have similar means
- Variables have similar variances

These assumptions come from the Euclidean distance

Dealing with skewed variables

- For variables taking only positive values: apply logarithmic transformation [Since logarithm is only defined for positive numbers, you can't take the logarithm of negative values]
- For variables with negative values: add a constant / calculate cubic root / Box-Cox transform

How does the Curse of Dimensionality affect k-Means Clustering?

- These plots show how the ratio of the standard deviation to the mean of distance between examples decreases as the number of dimensions increases.
- This convergence means k-means becomes less effective at distinguishing between examples.
- This negative consequence of high-dimensional data is called the curse of dimensionality.

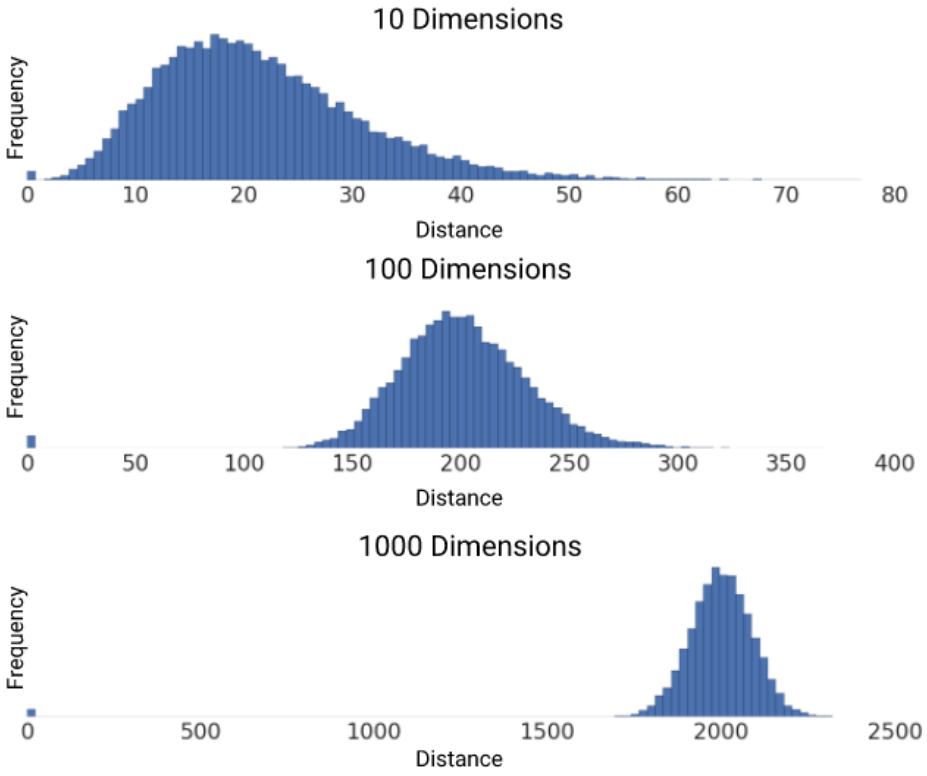


Figure 3: A demonstration of the curse of dimensionality. Each plot shows the pairwise distances between 200 random points.

As we can observe, when we have less dimension distance range between two data points have a good range [0-50], but as we increase the number of dimensions distance range between two data points decreases as there are multiple dimensions there overall distance between two data points over different hyperspace have lower distance range. As we have 1000 dimensions, distance range between two data points are around 2000 only. So we need to decrease the dimension in order to get separation or clustering in the datapoints.

Spectral clustering avoids the curse of dimensionality by adding a pre-clustering step to your algorithm:

1. Reduce the dimensionality of feature data by using PCA.
2. Project all data points into the lower-dimensional subspace.
3. Cluster the data in this subspace by using your chosen algorithm.