Initially done with Number of clusters as 2 , K =2

0 back. 2203

0 buffer\_overflow. 30

0 ftp\_write. 8

0 guess\_passwd. 53

0 imap. 12

0 ipsweep. 12481

0 land. 21

0 loadmodule. 9

0 multihop. 7

0 neptune. 1072017

0 nmap. 2316

0 normal. 972781

0 perl. 3

0 phf. 4

0 pod. 264

0 portsweep. 10412

0 rootkit. 10

0 satan. 15892

0 smurf. 2807886

0 spy. 2

0 teardrop. 979

0 warezclient. 1020

0 warezmaster. 20

1 portsweep. 1

It’s clear from the above output that only one sample is in the cluster 1 and rest are in cluster 0.

The conclusion which is drawn from the above experiments that two clusters are plainly insufficient.

How to choose K?

There are 23 different output class in data, so at least K =23 could be the minimum number of cluster required. But for this data this is not the right answer.

What is a good cluster?

A clustering can be considered good if each data point were near to its closest centroid. So we define a technique to get the clustering score based on number of clusters formed by measuring the distance of all the points from its closest centroid. We performed this experiment from number 5 to 40 to see the clustering score in different scenario.

(5,1938.8583418059206)

(10,1629.4697800260826)

(15,1406.596097363899)

(20,1094.610764459)

(25,933.0761383541766)

(30,937.716926129182)

(35,788.5376325929751)

(40,433.7254427534861)

One thing is very obvious in clustering is that , as more clusters are added ,it should always be possible to make data points closer to a nearest centroid. But from above results its not helping much in cluster formation when we move fron number 25 to 30 the score actually increases which violates the main assumption of clustering which clearly shows that we are still not at the correct number of K.

What could be the problem above?

The random starting set of clusters chosen for K =30 perhaps led to particularly suboptimal clustering or, it may have stopped early before it reached to a local optimum.

How can we improve this?

We can improve this by running the clustering many times for a values of k, with random staring centroid set each time and picking the best clustering . We have option for setting number of runs. Also the algorithm has threshold via setEpsilon that controls the minimum amount cluster centroid movement that is considered significant, lower values means the K-means algorithm will let the centroid continue to move longer.

K =10 and epsilon = 1.0 e-6

(30,863.1178555413134)

(40,771.7590035982141)

(50,334.8217659498883)

(60,293.88754477795726)

(70,291.71265062381804)

(80,139.80730239883218)

(90,148.1175126947914)

(100,116.26461248020944)

Here it seems decreasing notably past 100. So right value of K must be past 100.

Sample data representation in R:

Feature Normalization:

Running on different value of K: