Homework - 4

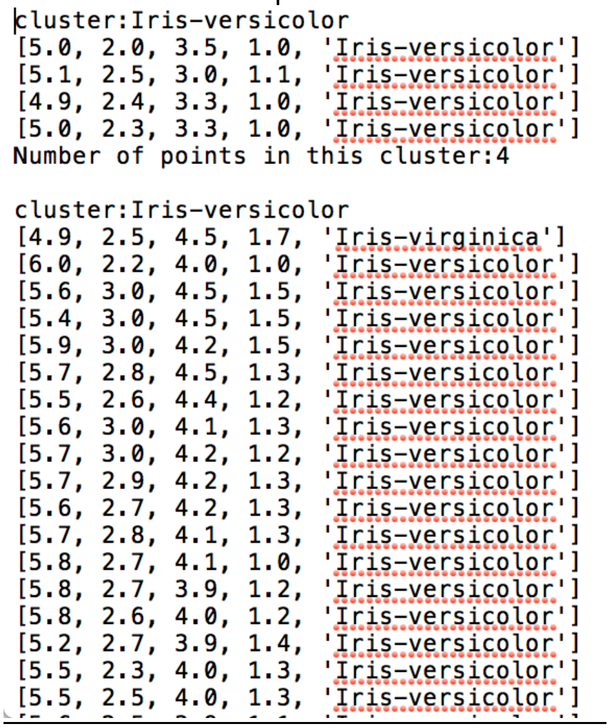
Contents of the folder -
1) Input (iris.data.txt)
2) Output (SaiSree_Kamineni_Output_k.txt)
3) Solution – Scala code (SaiSree_Kamineni_clustering.scala), Jar files (SaiSree_Kamineni_clustering.jar)
4) Description File (SaiSree_Kamineni_description.pdf)

Note - I haven't set up SPARK_HOME variable. So I run the scripts and commands from inside "*spark-1.6.3-bin-hadoop2.4*" directory.
Place this UnZipped folder in *spark-1.6.3-bin-hadoop2.4* directory

Steps to run the jar file -
./bin/spark-submit --class "HW4" --master local[4] ./SaiSree_Kamineni_hw4/Solution/SaiSree_Kamineni_clustering.jar "./SaiSree_Kamineni_hw4/Input/iris.data.txt" k

Output will be stored in current directory that is spark-1.6.3-bin-hadoop2.4 with name SaiSree_Kamineni_Output_k.txt

Output format
As mentioned in the problem statement. For k=4



```
cluster:Iris-versicolor
[5.0, 2.0, 3.5, 1.0, 'Iris-versicolor']
[5.1, 2.5, 3.0, 1.1, 'Iris-versicolor']
[4.9, 2.4, 3.3, 1.0, 'Iris-versicolor']
[5.0, 2.3, 3.3, 1.0, 'Iris-versicolor']
Number of points in this cluster:4

cluster:Iris-versicolor
[4.9, 2.5, 4.5, 1.7, 'Iris-virginica']
[6.0, 2.2, 4.0, 1.0, 'Iris-versicolor']
[5.6, 3.0, 4.5, 1.5, 'Iris-versicolor']
[5.4, 3.0, 4.5, 1.5, 'Iris-versicolor']
[5.9, 3.0, 4.2, 1.5, 'Iris-versicolor']
[5.7, 2.8, 4.5, 1.3, 'Iris-versicolor']
[5.5, 2.6, 4.4, 1.2, 'Iris-versicolor']
[5.6, 3.0, 4.1, 1.3, 'Iris-versicolor']
[5.7, 3.0, 4.2, 1.2, 'Iris-versicolor']
[5.7, 2.9, 4.2, 1.3, 'Iris-versicolor']
[5.6, 2.7, 4.2, 1.3, 'Iris-versicolor']
[5.7, 2.8, 4.1, 1.3, 'Iris-versicolor']
[5.8, 2.7, 4.1, 1.0, 'Iris-versicolor']
[5.8, 2.7, 3.9, 1.2, 'Iris-versicolor']
[5.8, 2.6, 4.0, 1.2, 'Iris-versicolor']
[5.2, 2.7, 3.9, 1.4, 'Iris-versicolor']
[5.5, 2.3, 4.0, 1.3, 'Iris-versicolor']
[5.5, 2.5, 4.0, 1.3, 'Iris-versicolor']
```

Approach –

1) Defined distance method to calculate Euclidean distance & diff function to return the distance between two clusters.
2) Read input file and placed in array.
3) For every pair combination of points, calculate distance and added it to Priority Queue (Min heap).
4) Removed the min of queue and deleted the clusters in the min node from available clusters and added a new cluster combining clusters in min node.
5) Repeat 3,4 steps till number nodes > k
6) Copy all the required details to string and write to file.