|  |  |
| --- | --- |
| Lab 5 – Apache Spark | Name:ID: |

1. After the previous labs, you have learned how to implement k-means.using Python.
2. In this lab, you are asked to implement k-means using Scala commonly found in Apache Spark.
3. You are given the k-means template file “kmeans.scala”.
4. In the file, there are several spark functions to be implemented by yourself.
5. The following table summarizes the meanings of those functions.

|  |  |
| --- | --- |
| **Function** | **Description** |
| def distance(p:Vector[Double], q:Vector[Double]) : Double | It calculates the distance between two points “p” and “q”. |
| def clostestpoint(q: Vector[Double], candidates: Array[Vector[Double]]): Vector[Double] | Given a query point “q”, it finds the nearest point among “candidates”. |
| def add\_vec(v1: Vector[Double], v2: Vector[Double]): Vector[Double] = | It performs the addition of two points “v1” and “v2”. |
| def average(cluster: Iterable[Vector[Double]]): Vector[Double] | It finds the centroid of “cluster”. |

1. You are asked to code and fill in the content of each function.
2. After that, you are asked to use the functions to implement k-means using Scala.
3. Once you have finished the above coding, you are asked to code and run your k-means on the given data file “clustering\_dataset.txt” with k=3.
4. Please report the 3 cluster centroids you have found in the below table.

|  |  |
| --- | --- |
|  | **Data Values** |
| Centroid 1 |  |
| Centroid 2 |  |
| Centroid 3 |  |

1. Please upload your “kmeans.scala” to the submission system.
2. Please also upload this sheet with your answers to the submission system.
3. This is the end.