Using Mahout Library for Clustering Algorithm: A Case Study on Healthcare Data

Dr. Divya Chauhan[1]; Dr. Satpal[2]

[1,2]Department of Computer Science, Department of Economics Government College, Shimla, Himachal Pradesh, India

Publication Date: 2025/12/03

Abstract: Data mining techniques and algorithms worked excellently with small datasets. Data mining algorithms analysed bulk data to identify trends and draw conclusions. But most data mining tool is not efficient to process very large dataset which is the case in big data. They are not able to give quick outcomes in quick time, unless the computational tasks are run on multiple machines distributed over cloud. For process large volume of data like big data, Hadoop has adopted a new set of library for machine learning called Mahout.

This paper deals with the clustering algorithms with the help of mahout library in Hadoop MapReduce environment. The real-world healthcare dataset is used which is quite large in size. The three clustering algorithms used are canopy clustering, K-Means clustering and fuzzy K-Means clustering.

Keywords: *Data Mining, Big Data, Hadoop, Mahout, Clustering, Healthcare.*

How to Cite: Dr. Divya Chauhan; Dr. Satpal (2025). Using Mahout Library for Clustering Algorithm: A Case Study on Healthcare Data. *International Journal of Innovative Science and Research Technology*, 10(11), 2344-2353.
https://doi.org/10.38124/ijisrt/25nov1348

I. INTRODUCTION

Healthcare organizations are generating a huge tsunami of data which requires efficient tools to analyse it and get useful information and patterns out of it. However the use of this data to the best of its abilities however remains a dream. By adopting the right analytics solution, healthcare providers can make better and more meaningful decisions based on evidence and insights derived from their data. Big data analytics can be good to analyse big data and improve results by applying advanced analytical techniques and discover hidden insights [1].

For process large volume of data like big data, Hadoop has adopted a new set of library for machine learning called Mahout. Mahout allows breaking down a computation task into multiple segments and run each segment on a different machine. Mahout is an open-source library that usually runs coupled with the Hadoop infrastructure to manage huge volumes of data efficiently.

- *Mahout*

Apache Mahout [2] is an open-source project that is primarily used for creating scalable machine learning algorithms for handling large datasets. It runs on Hadoop, using the MapReduce paradigm. Mahout requires Java 7 or above to be installed, and also needs a Hadoop, Spark, H2O, or Flink platform for distributed processing Machine learning is a discipline of artificial intelligence that enables systems to learn based on data alone, continuously improving performance as more data is processed. Mahout offers the coder a ready-to-use framework for doing data mining tasks on large volumes of data. While Mahout has only been around for a few years, it has established its place in the field of machine learning technologies. Popular organizations such as LinkedIn, Twitter, Foursquare, Adobe, Facebook, and Yahoo use Mahout internally. Foursquare helps individual to find out places, food, and entertainment available in a particular area. It uses the recommender engine of Mahout. Twitter uses Mahout for user interest modelling. Yahoo! uses Mahout for pattern mining. There are many machine learning algorithms that are exposed by Mahout. Collaborative Filtering has Item-based Collaborative Filtering,Matrix Factorization with Alternating Least Squares, Matrix Factorization with Alternating Least Squares on Implicit Feedback. Classification can be implemented using Naive Bayes, Complementary Naive Bayes, Random Forest. Clustering has five algorithms Canopy Clustering, k-Means Clustering, Fuzzy k-Means, Streaming k-Means. Dimensionality Reduction is implemented using Lanczos Algorithm, Stochastic SVD, principal Component Analysis. And other machine learning algorithms like Frequent Pattern Matching foe association, spectral clustering, and row similarity job to name a few.

➢ *Clustering*

A cluster refers to a small group of objects [3]. Clustering means grouping any forms of data like text files, into characteristically similar groups. In other words, clustering distributes data points into homogeneous clusters, such that the points in the same group are as related as possible, while those in different groups are as dissimilar as possible. When a collection of points is given, they are divided into groups based on similarity criterion.

Good clustering techniques will produce a good or a high-quality cluster [4]. Clustering groups data instances into subsets in such a manner that similar instances are grouped together, while different instances are distributed to different groups and these groups are called as clusters. [5]. Clustering techniques are used in many fields such as market research in image processing for segmentation of images and pattern recognition, to discover different groups of customers, and in text or document classification for information discovery. These techniques are also used in outlier detection which means identifying observations that do not belong to any cluster in particular, to help in identifying fraud in online transactions, etc.

There are three main types of clustering which are used in this paper. They are: Canopy clustering, K-Means clustering, fuzzy K-Means clustering.

- *Canopy Clustering*:

It is a very simple, fast and surprisingly accurate method for grouping points into clusters. All points are represented as a point in a multidimensional feature space. Canopy Clustering is often used as an initial step in more rigorous clustering techniques, such as K-Means Clustering to use cluster information as random seeds. By getting starting with initial clustering points, the distance measurements can be significantly reduced by ignoring points outside of the initial canopies. The algorithm uses a fast approximate distance metric and two distance thresholds t1 and t2 for processing where t1>t2.

The algorithm begins with the data points being clustered together. Then it begins to form a new canopy by removing one point from the set. If the distance to the first point of canopy is less than t1, a new canopy is assigned to the point. Further if the distance of point is less than t2 it is removed from the original set. The process is repeated until there is no data point left in the set to cluster.

- *K-Means Clustering:*

It was discovered by Macqueen in 1967, is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K-Means clustering is a method of vector quantization, which originally comes from signal processing, a popular technique for cluster analysis. The purpose of k in K-Means clustering is to divide the data into k non-empty subsets. It then assigns each point to a specific cluster. Next step is to find the distance of each point from the centroid and allot points to the cluster where the distance of each point from the centroid is minimum. The same process is continued for number of iteration specified an d final centroids of the new clusters are formed

- *Fuzzy K-Means Clustering:*

Unlike k-Means clustering fuzzy clustering generalizes partition clustering methods by allowing a point to be partially classified into more than one cluster. In hard clustering each point is a member of only one cluster. It is much easier to create fuzzy boundaries than it is to settle a point for a single cluster. One of the most difficult required skills in cluster analysis is to choose the appropriate number of clusters for the dataset being processed. The degree of fuzziness in a model may be measured by Dunn’s partition coefficient which measures how close the fuzzy solution is to the corresponding hard solution. This hard solution is formed by classifying each point into the cluster which has the largest membership value. The procedure is to randomly select the number of clusters to be formed according to the dataset being used. The centroid is calculated with the help of parameters like fuzzy membership, fuzziness parameter and data points. The distance of each point is calculated from the centroid and the value of membership parameter is updated accordingly. The process is repeated until a constant value is obtained for the membership parameter.

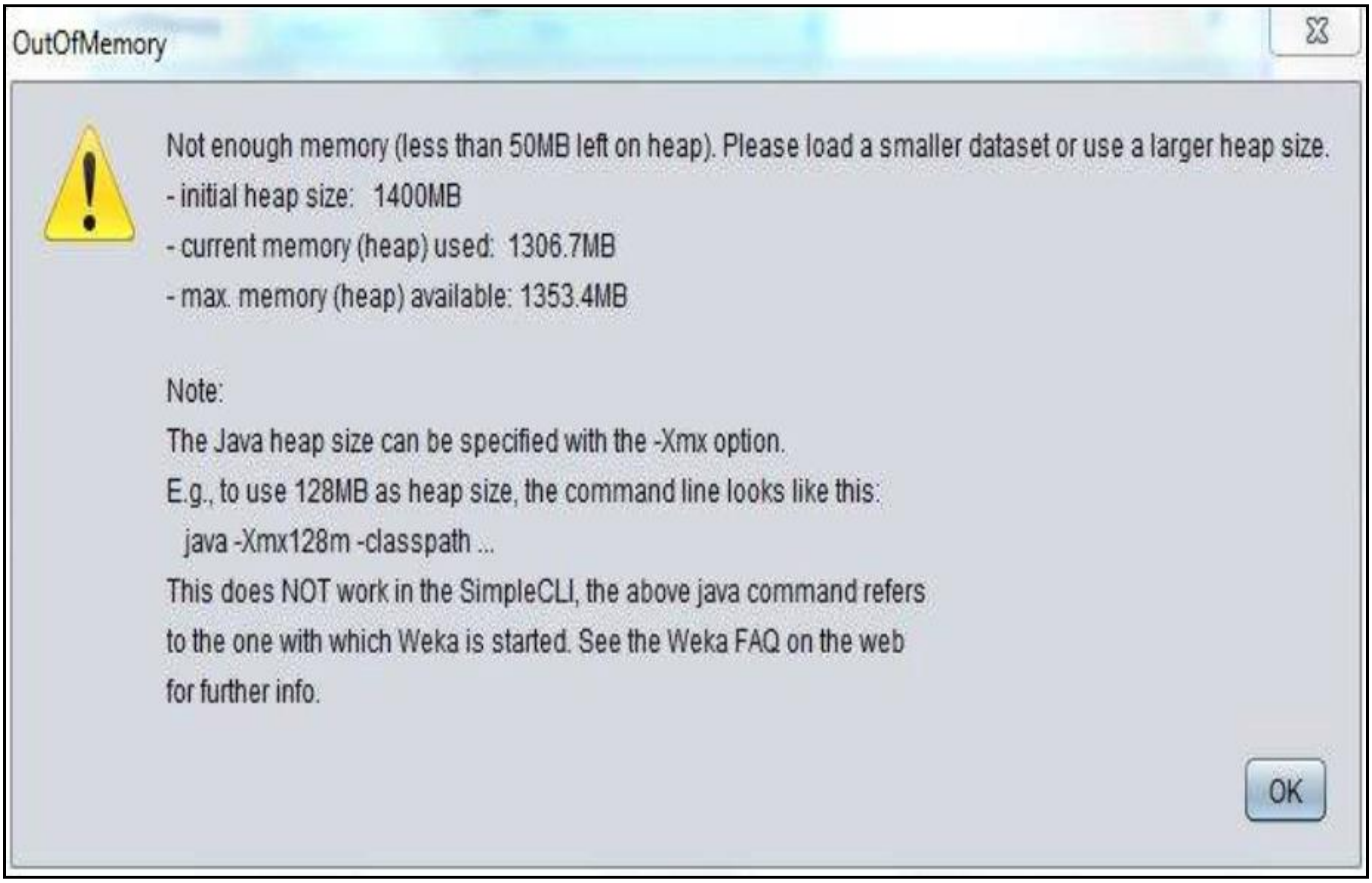
II. LITERATURE REVIEW

Dr. Venkateswara Reddy Eluri [6] et al used three clustering algorithms on mahout namely K-means, Fuzzy K-Means (FKM) and Canopy clustering and compared them. The result showed that K-Means clustering algorithm is suitable for globular data set but not for non globular data set and also concluded that identifying the number of clusters initially is difficult for big data set. Fatos Xhafa [7] et al analysed the performance of clustering algorithms of Apache Mahout using a Twitter streaming dataset under a Hadoop MapReduce cluster infrastructure according to various evaluation criteria. It was observed that significant reduction in processing time can be achieved for clustering algorithms executed on Hadoop MapReduce. Van-Dai Ta [8] et al proposed a generic architecture for big data healthcare analytic by using open sources, including Hadoop, Apache Storm, Kafka and NoSQL Cassandra. For future more power tools for data analytics such as machine learning are recommended to gain efficiency. Rui Máximo Esteves and Chunming Rong [9] compared k-means and fuzzy c-means for clustering a noisy realistic and big dataset using a free cloud computing solution Apache Mahout/ Hadoop and Wikipedia’s latest articles. It was observed that in a noisy dataset, fuzzy c-means can lead to worse cluster quality than k-means and the execution times of both algorithms have high variances according to the initial seeding. It was also found that that Mahout is a promise clustering technology but the pre-processing tools are not developed enough for an efficient dimensionality reduction. Ahmad Al-Khoder, Hazar Harmouch [10] compared four data mining tools by using three classifier algorithms namely Naïve Bayes (NB), Decision Tree (DT), and K

Nearest Neighbor (KNN). Tools are compared against five criteria namely: platform, input/output formats, performance, visualization, popularity, structure and development. Following observations were highlighted: R seems to support wider range of input/output formats, and visualization types. WEKA was the best tool to run the selected classifiers followed by R, RapidMiner, and finally KNIME. Hoda A. Abdel Hafez [11] reviewed big data mining, its challenges, different techniques and open sources tools for handling large data sets. A discussion was made to implement all these to telecommunication and the benefits and opportunities gained from them. Amineh Amini [12] et al overviewed the density-based data stream clustering algorithms and the evaluation metrics. Many research directions were highlighted. Such as developing clustering algorithms with fewer parameters and can handle various types of data streams such as categorical or uncertain data, making hybrid of different algorithms, evaluating algorithm on real life datasets. Olga Kurasova [13] et al. analysed challenges and problems of big data clustering with brief discussions of clustering algorithms and methods. Pritika Talwar [14] et.al. reviewed various clustering algorithms addressing different data types and objectives. The wide range applications of clustering were explored including the challenges. A.A. Wani [15] analyzed the strengths and limitations of five primary clustering algorithms. The paper highlighted the area where the future efforts to be put and suggestions for integration with emerging technologies like deep learning and quantum computing.

III. RESULTS AND ANALYSIS

Data mining techniques works perfectly when the size of data is small. But now data is money and every organization is generating and using it in abundance. It has been observed that Weka cannot handle very large datasets because it supports only sequential single node execution [16]. Hence, the size of datasets and processing tasks that Weka can handle within its existing environment is limited both by the amount of memory in a single node and by sequential execution. To overcome this limitation and for handling large datasets, parallel and distributed computing-based systems and technologies are required. Hadoop based technologies and associated libraries are the most popular solution for handling large datasets. Figure 1 show the error message when the large dataset is processed in Weka explorer environment and suggested to increase the memory with usage of Simple CLI.

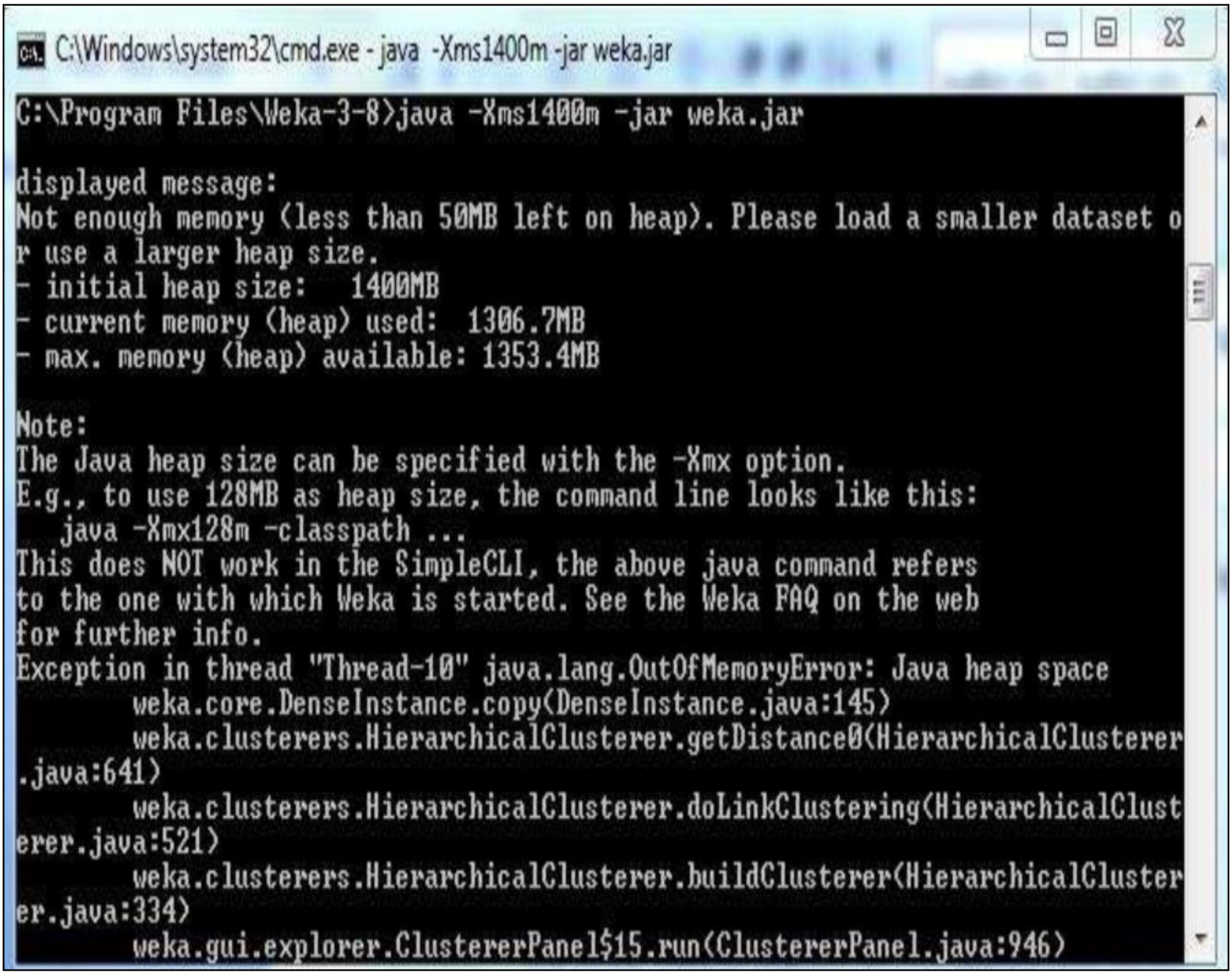

OutOfMemory

Not enough memory (less than 50MB left on heap). Please load a smaller dataset or use a larger heap size.
- initial heap size: 1400MB
- current memory (heap) used: 1306.7MB
- max. memory (heap) available: 1353.4MB

Note:
The Java heap size can be specified with the -Xmx option.
E.g., to use 128MB as heap size, the command line looks like this:
java -Xmx128m -classpath ...
This does NOT work in the SimpleCLI, the above java command refers
to the one with which Weka is started. See the Weka FAQ on the web
for further info.

Fig 1 Error Message of Weka Explorer for Big Data [14]

Even after using Simple CLI as suggested in error message, Weka could not process the data as shown in figure 2. One of reasons could be that many data mining tools load complete data into RAM to analyse it. Hadoop framework gives the privilege to store the data in its distributed file system called HDFS.



```
C:\Program Files\Weka-3-8>java -Xms1400m -jar weka.jar

displayed message:
Not enough memory (less than 50MB left on heap). Please load a smaller dataset o
r use a larger heap size.
- initial heap size:   1400MB
- current memory (heap) used:  1306.7MB
- max. memory (heap) available: 1353.4MB

Note:
The Java heap size can be specified with the -Xmx option.
E.g., to use 128MB as heap size, the command line looks like this:
   java -Xmx128m -classpath ...
This does NOT work in the SimpleCLI, the above java command refers
to the one with which Weka is started. See the Weka FAQ on the web
for further info.
Exception in thread "Thread-10" java.lang.OutOfMemoryError: Java heap space
        weka.core.DenseInstance.copy(DenseInstance.java:145)
        weka.clusterers.HierarchicalClusterer.getDistance0(HierarchicalClusterer
.java:641)
        weka.clusterers.HierarchicalClusterer.doLinkClustering(HierarchicalClust
erer.java:521)
        weka.clusterers.HierarchicalClusterer.buildClusterer(HierarchicalCluster
er.java:334)
        weka.gui.explorer.ClustererPanel$15.run(ClustererPanel.java:946)
```

Fig 2 Error Message of Weka Simple CLI [14]

➢ *Experimental Setup*

The dataset collected are pdf files from various online sites of hospitals like AIIMS [17], TMC [18], Fortis [19], Apollo [20] to name a few. The annual report generated by various hospitals in India for last ten years is downloaded. The pdf files are converted into text files so that it can be converted into sequential file which is understood by Mahout.

The data collected is large in size so it is stored in HDFS with distributed multimode Hadoop cluster. Hadoop distributive cluster setup is acquired to increase the scalability of the architecture. Five computers are connected to each other amongst which one acts as master and other four as slaves. The master node contains the name node which comprises of job tracker in order to assign tasks to the data nodes whereas slave nodes consist of the Datanode where the actual data is stored and task tracker in each Datanode takes care of it. The software configuration of computers connected in the cluster is as follow: 64-bit computers with 4 GB RAM, 1TB hard disk, operating system Ubuntu, Core i3 processors.

- *Poppler-Utils [21] of Ubuntu is Used for the Conversion of Pdf File into Text File Format Using Command:*

✓ *Sudopdftotext<Pdf_File><Text_File>*

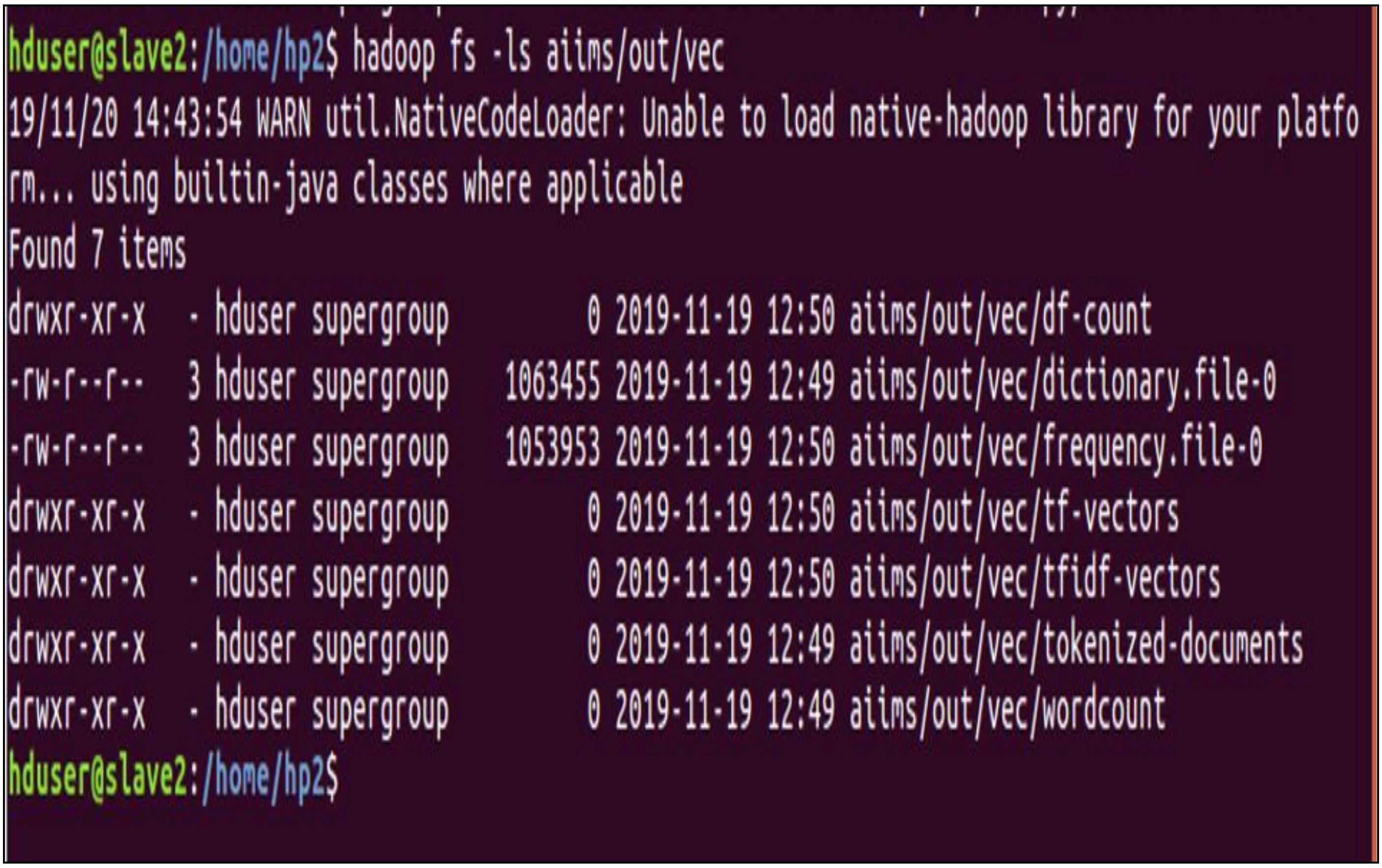
The text file is to be converted into sequence file format which can be processed by Mahout. Mahout provides a utility to convert given input file into a sequence file format. The syntaxfor conversion is:

✓ *Seqdirectory –I<Input_File> -O <Ouput_File> -Ow –Method <Mapreduce> -Chunk <Chunk Size>*

The parameter input_file is the directory where the input files are stored that the text files. The other parameter output_file is the path to output file directory where the sequence file will be generated. –ow signifies to overwrite the file if it is already created. The sequence file will consist of Part-r-00000 file. This file is further to be converted into vector form. To generate vector files from sequence file format, Mahout provides the seq2sparse utility. The syntax is as:

- ✓ *Seq2sparse –I<Input_File> -O <Output_File> -A <Analyzername> -Wt<Weight_Name> -Chunk<Chunk_Size>*

Here the Input_file denotes the filepath to sequence directory and output_file denotes the path where the vector file is to be stored. The weight_name is usually tf or tfidf format. Format selected is tdidf for this work. Chunk size is specified in MB. The output generated from this command is a vector directory, a dictionary file, frequency file, wordcount directory as shown in figure 3 below.



```
hduser@slave2:/home/hp2$ hadoop fs -ls aiims/out/vec
19/11/20 14:43:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
Found 7 items
drwxr-xr-x   - hduser supergroup          0 2019-11-19 12:50 aiims/out/vec/df-count
-rw-r--r--   3 hduser supergroup    1063455 2019-11-19 12:49 aiims/out/vec/dictionary.file-0
-rw-r--r--   3 hduser supergroup    1053953 2019-11-19 12:50 aiims/out/vec/frequency.file-0
drwxr-xr-x   - hduser supergroup          0 2019-11-19 12:50 aiims/out/vec/tf-vectors
drwxr-xr-x   - hduser supergroup          0 2019-11-19 12:50 aiims/out/vec/tfidf-vectors
drwxr-xr-x   - hduser supergroup          0 2019-11-19 12:49 aiims/out/vec/tokenized-documents
drwxr-xr-x   - hduser supergroup          0 2019-11-19 12:49 aiims/out/vec/wordcount
hduser@slave2:/home/hp2$
```

Fig 3 Conversion of Sequence File to Vector Files

The *dictionary*. file-0 file contains the mapping between a term and its integer ID, and the other folders are intermediate folders generated during the vectorization process. TF-IDF stands for term frequency-inverse document frequency. The number of times a term occurs in a document is called its term frequency. Term frequency is calculated as the ratio of the number of times the word occurs in the document to the total number of words in the document. tf*idf is a multiplication of tf and idf where idf is the inverse document frequency which is the log of the ratio of total number of documents and number of documents containing the term.

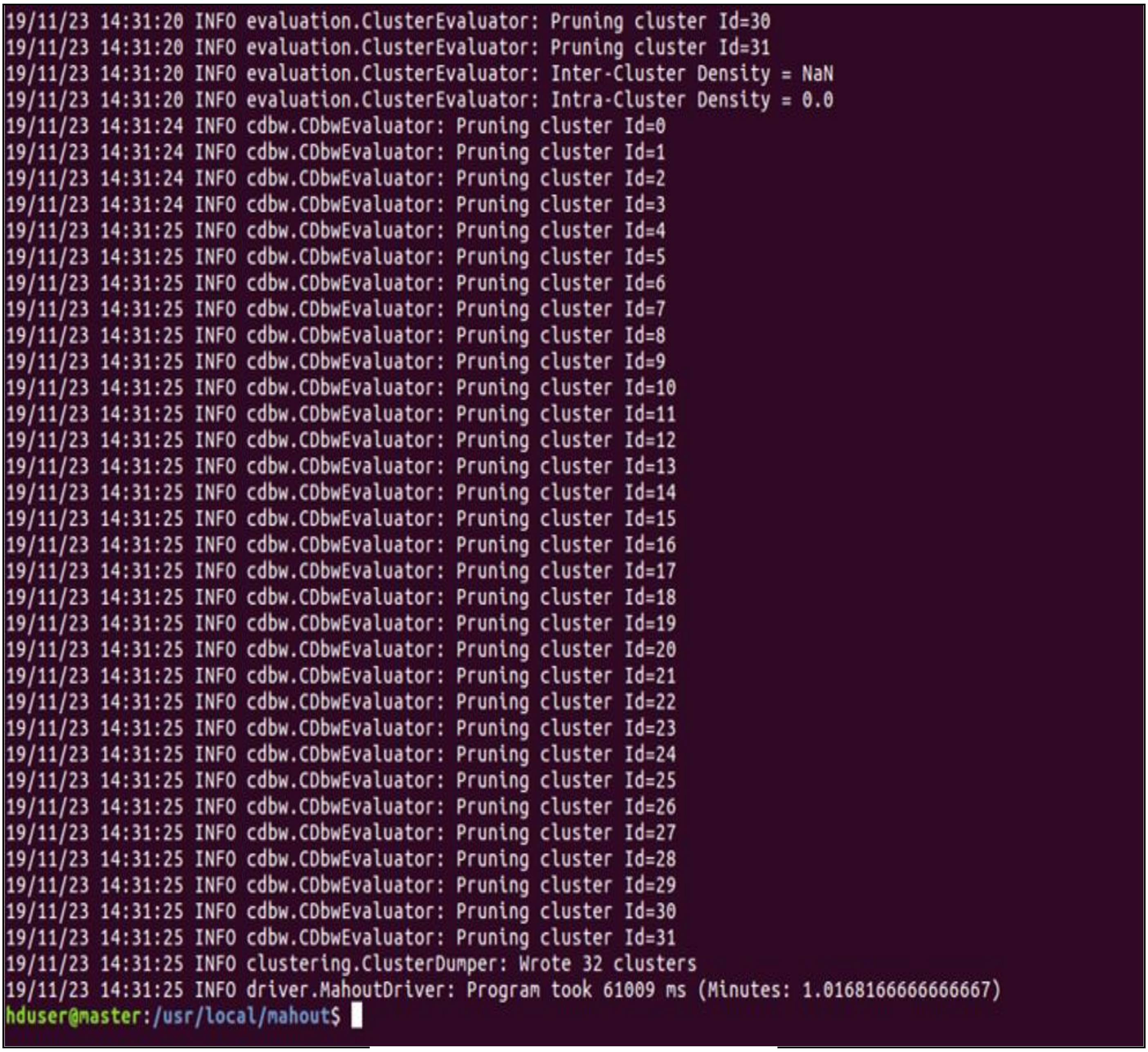
➢ *Analysis*

As already described above canopy clustering is a simple and fast technique used by Mahout for clustering purpose. This clustering algorithm is often used as an initial step in other clustering techniques such as k-means clustering. After converting the data into required vector form each mapper perform canopy clustering on data and output its canopy centers. The task of reduces is to cluster the canopy centers to produce the final canopy centers. At the end final canopies are produced with relevant points associated with them. The mahout command for canopy clustering is as follows:

- *Canopy –I<Input_File> -O<Output_File> -T1<Value> -T2<Value> -Dm<Distance_Measure> -Cl*

Here input_file is the path to tfidf-vector file and output_file is the directory path where the cluster information and cluster points is to be stored. The distance measure used in for clustering is Euclidean distance measure.

After the dumping the cluster into clusterdump utility of mahout certain information can be obtained as shown in in figure 4 below. The number of clusters formed by canopy is 32 with intra cluster density of 0.0. The cluster information can also be fed to other clustering algorithms to act as random seeds.



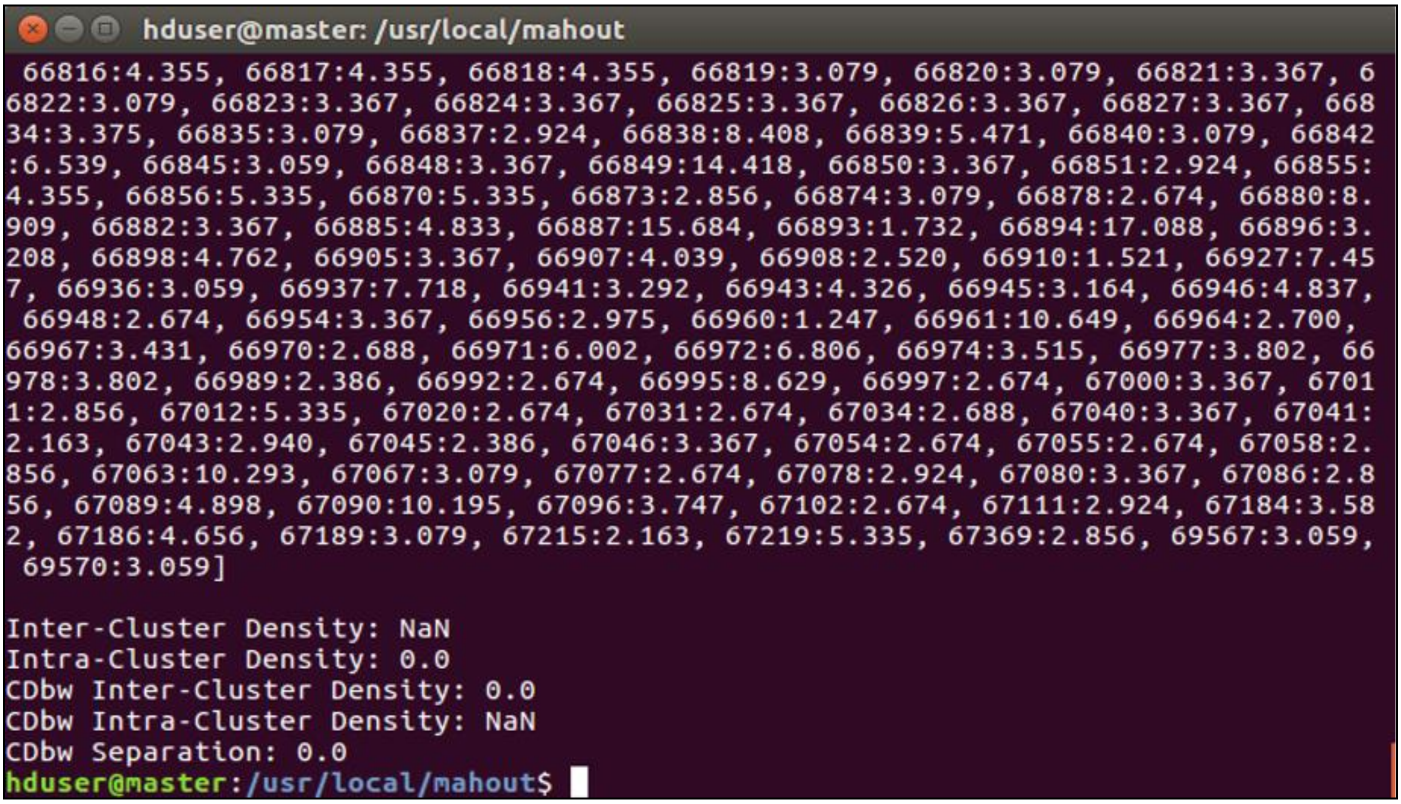
```
19/11/23 14:31:20 INFO evaluation.ClusterEvaluator: Pruning cluster Id=30
19/11/23 14:31:20 INFO evaluation.ClusterEvaluator: Pruning cluster Id=31
19/11/23 14:31:20 INFO evaluation.ClusterEvaluator: Inter-Cluster Density = NaN
19/11/23 14:31:20 INFO evaluation.ClusterEvaluator: Intra-Cluster Density = 0.0
19/11/23 14:31:24 INFO cdbw.CDbwEvaluator: Pruning cluster Id=0
19/11/23 14:31:24 INFO cdbw.CDbwEvaluator: Pruning cluster Id=1
19/11/23 14:31:24 INFO cdbw.CDbwEvaluator: Pruning cluster Id=2
19/11/23 14:31:24 INFO cdbw.CDbwEvaluator: Pruning cluster Id=3
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=4
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=5
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=6
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=7
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=8
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=9
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=10
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=11
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=12
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=13
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=14
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=15
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=16
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=17
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=18
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=19
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=20
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=21
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=22
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=23
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=24
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=25
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=26
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=27
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=28
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=29
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=30
19/11/23 14:31:25 INFO cdbw.CDbwEvaluator: Pruning cluster Id=31
19/11/23 14:31:25 INFO clustering.ClusterDumper: Wrote 32 clusters
19/11/23 14:31:25 INFO driver.MahoutDriver: Program took 61009 ms (Minutes: 1.0168166666666667)
hduser@master:/usr/local/mahout$
```

Fig 4 Cluster Evaluation of Canopy Clustering

Two Cases are formed to implement k_Means clustering: First case takes clusters formed by canopy algorithm as initial random seeds for cluster formation. Second case takes k value. The k in k-means clustering algorithm represents the number of clusters the data is to be divided into. Here the number of clusters chosen are 10 with maximum iteration of 10. There are several algorithms for the distance measure and the one adopted here is Euclidean Distance measure. K-means clustering job uses vector directory as input, clusters directory, distance measure, maximum number of iterations to be carried out, and an integer value representing the number of clusters the input data is to be divided into. The utility provided by mahout for K-Means clustering is kmeans which is used as:

- *Kmeans –I<Input_File> -O <Output_File> -C<Cluster_File> -Dm<Distance_Measure> -X<Value>-K <Value>-Cl*

Here the input is the vector tfidf directory path, out I sthe path to directory where the clusterpoints and clusters are to be stored. X is the maximum number of iteration to be performed to calculate cluster centroids. C argument takes the empty directory to generate random seeds if number of clusters is provided otherwise the cluster directory created by canopy algorithm. Figure 5 shows the output of clusterdump for K-Means clustering using canopy clusterpoints as initial random seeds. Total 32 clusters are created by the algorithm with intra cluster density of 0 and CDbw separation of 0.0 approximately.

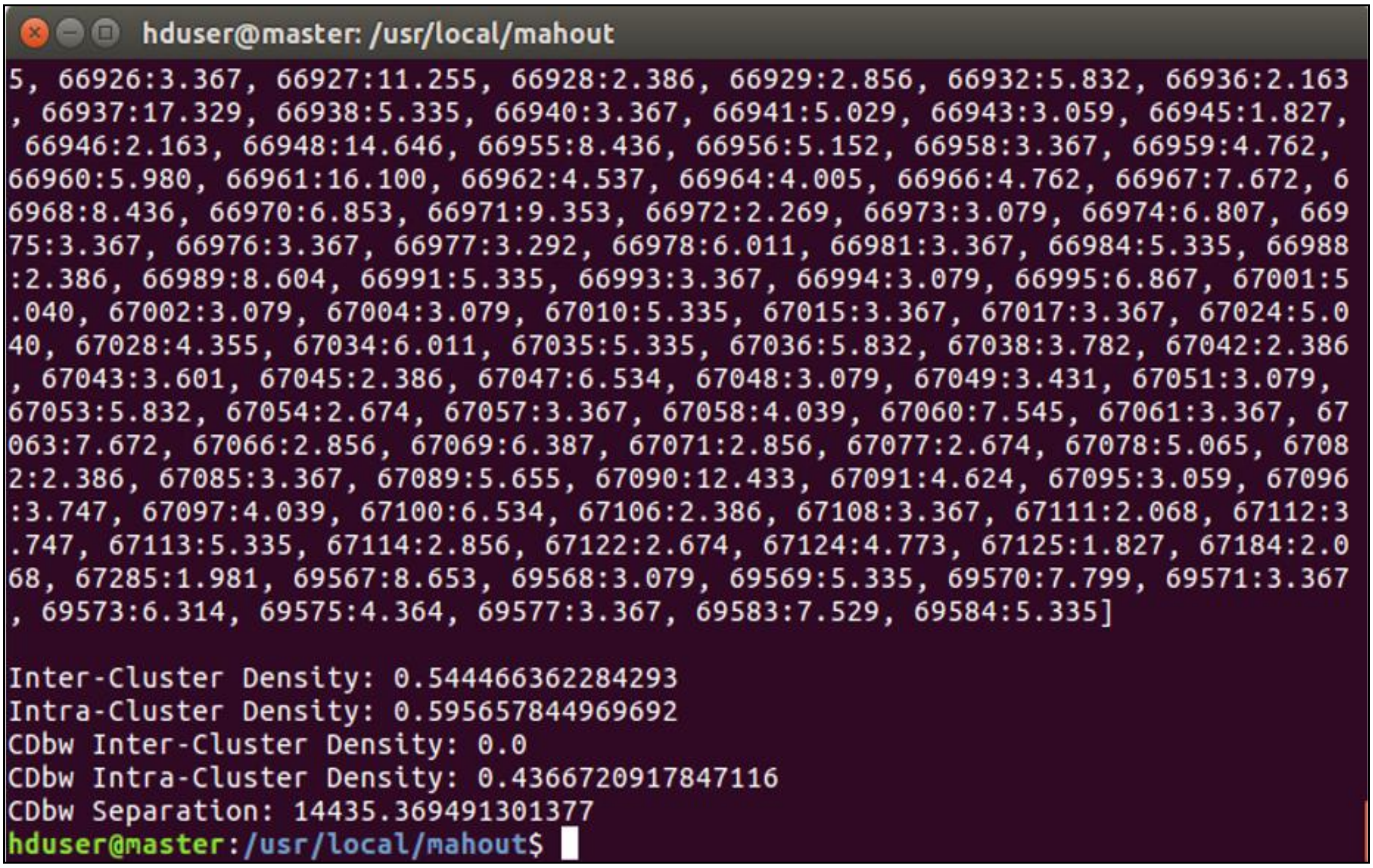


```
 66816:4.355, 66817:4.355, 66818:4.355, 66819:3.079, 66820:3.079, 66821:3.367, 6
6822:3.079, 66823:3.367, 66824:3.367, 66825:3.367, 66826:3.367, 66827:3.367, 668
34:3.375, 66835:3.079, 66837:2.924, 66838:8.408, 66839:5.471, 66840:3.079, 66842
:6.539, 66845:3.059, 66848:3.367, 66849:14.418, 66850:3.367, 66851:2.924, 66855:
4.355, 66856:5.335, 66870:5.335, 66873:2.856, 66874:3.079, 66878:2.674, 66880:8.
909, 66882:3.367, 66885:4.833, 66887:15.684, 66893:1.732, 66894:17.088, 66896:3.
208, 66898:4.762, 66905:3.367, 66907:4.039, 66908:2.520, 66910:1.521, 66927:7.45
7, 66936:3.059, 66937:7.718, 66941:3.292, 66943:4.326, 66945:3.164, 66946:4.837,
 66948:2.674, 66954:3.367, 66956:2.975, 66960:1.247, 66961:10.649, 66964:2.700,
66967:3.431, 66970:2.688, 66971:6.002, 66972:6.806, 66974:3.515, 66977:3.802, 66
978:3.802, 66989:2.386, 66992:2.674, 66995:8.629, 66997:2.674, 67000:3.367, 6701
1:2.856, 67012:5.335, 67020:2.674, 67031:2.674, 67034:2.688, 67040:3.367, 67041:
2.163, 67043:2.940, 67045:2.386, 67046:3.367, 67054:2.674, 67055:2.674, 67058:2.
856, 67063:10.293, 67067:3.079, 67077:2.674, 67078:2.924, 67080:3.367, 67086:2.8
56, 67089:4.898, 67090:10.195, 67096:3.747, 67102:2.674, 67111:2.924, 67184:3.58
2, 67186:4.656, 67189:3.079, 67215:2.163, 67219:5.335, 67369:2.856, 69567:3.059,
 69570:3.059]

Inter-Cluster Density: NaN
Intra-Cluster Density: 0.0
CDbw Inter-Cluster Density: 0.0
CDbw Intra-Cluster Density: NaN
CDbw Separation: 0.0
hduser@master:/usr/local/mahout$
```

Fig 5 Cluster Information of K-Means Clustering with Canopy

Figure 6 shows the cluster information of K-Means clustering where k value is set to 10. Ten clusters are formed with inter cluster density of 0.54, intra cluster density of 0.59 and CDbw separation of 14435 approx.



```
5, 66926:3.367, 66927:11.255, 66928:2.386, 66929:2.856, 66932:5.832, 66936:2.163
, 66937:17.329, 66938:5.335, 66940:3.367, 66941:5.029, 66943:3.059, 66945:1.827,
 66946:2.163, 66948:14.646, 66955:8.436, 66956:5.152, 66958:3.367, 66959:4.762,
66960:5.980, 66961:16.100, 66962:4.537, 66964:4.005, 66966:4.762, 66967:7.672, 6
6968:8.436, 66970:6.853, 66971:9.353, 66972:2.269, 66973:3.079, 66974:6.807, 669
75:3.367, 66976:3.367, 66977:3.292, 66978:6.011, 66981:3.367, 66984:5.335, 66988
:2.386, 66989:8.604, 66991:5.335, 66993:3.367, 66994:3.079, 66995:6.867, 67001:5
.040, 67002:3.079, 67004:3.079, 67010:5.335, 67015:3.367, 67017:3.367, 67024:5.0
40, 67028:4.355, 67034:6.011, 67035:5.335, 67036:5.832, 67038:3.782, 67042:2.386
, 67043:3.601, 67045:2.386, 67047:6.534, 67048:3.079, 67049:3.431, 67051:3.079,
67053:5.832, 67054:2.674, 67057:3.367, 67058:4.039, 67060:7.545, 67061:3.367, 67
063:7.672, 67066:2.856, 67069:6.387, 67071:2.856, 67077:2.674, 67078:5.065, 6708
2:2.386, 67085:3.367, 67089:5.655, 67090:12.433, 67091:4.624, 67095:3.059, 67096
:3.747, 67097:4.039, 67100:6.534, 67106:2.386, 67108:3.367, 67111:2.068, 67112:3
.747, 67113:5.335, 67114:2.856, 67122:2.674, 67124:4.773, 67125:1.827, 67184:2.0
68, 67285:1.981, 69567:8.653, 69568:3.079, 69569:5.335, 69570:7.799, 69571:3.367
, 69573:6.314, 69575:4.364, 69577:3.367, 69583:7.529, 69584:5.335]

Inter-Cluster Density: 0.544466362284293
Intra-Cluster Density: 0.595657844969692
CDbw Inter-Cluster Density: 0.0
CDbw Intra-Cluster Density: 0.4366720917847116
CDbw Separation: 14435.369491301377
hduser@master:/usr/local/mahout$
```

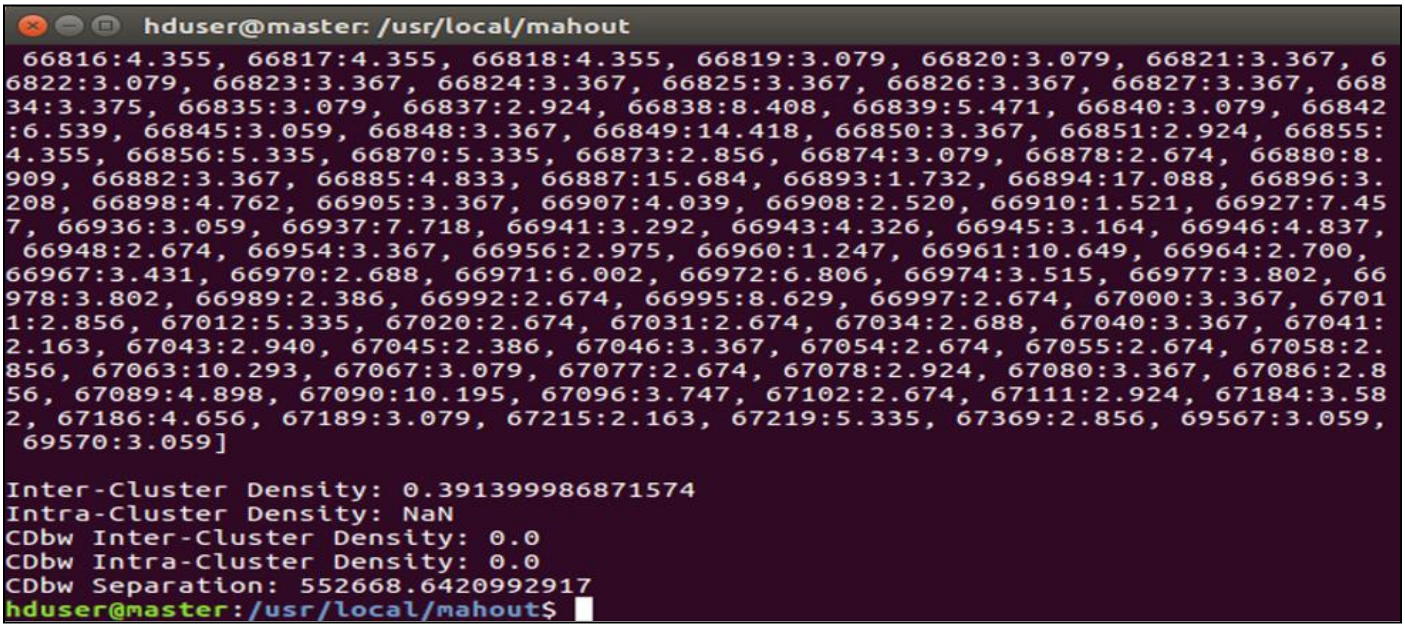
Fig 6 K-Means Clustering for 10 Clusters.

Fuzzy clustering is a soft clustering algorithm where data points belongs to one or more clusters and their membership in a particular cluster corresponding to some probability.The number of reducer tasks has set equal to the map tasks. During each iteration a new file is created which contains the modified cluster centers. A map task is run to output the points and cluster membership to each cluster as final output to a directory named points. The utility provided by mahout for fuzzy K-Means Clustering is fkmeans.

- *Fkmeans –I<Input> -O<Output> -C<Cluster Directory>-M<Fuzziness_Argument>-X<Value> -K<Value> -Dm<Distance Measure>-E –Cl*

Input is the path to vector directory, output is the path to store the output of clustering with information of cluster points, fuzziness argument should be greater than one which signifies the degree of normalization, -x is the maximum number of iteration to be performed on the data, -e signifies the property to emit vector to most likely cluster during clustering.

Like above two cases are created for fuzzy K-means clustering. Figure 7 shows the output of fuzzy kmeans clustering after using clusterdump utility of mahout with canopy fed as initial clusters. Total 32 clusters are formed. The inter cluster density of clusters is 0.39 and intra cluster density is 552668 approximately.

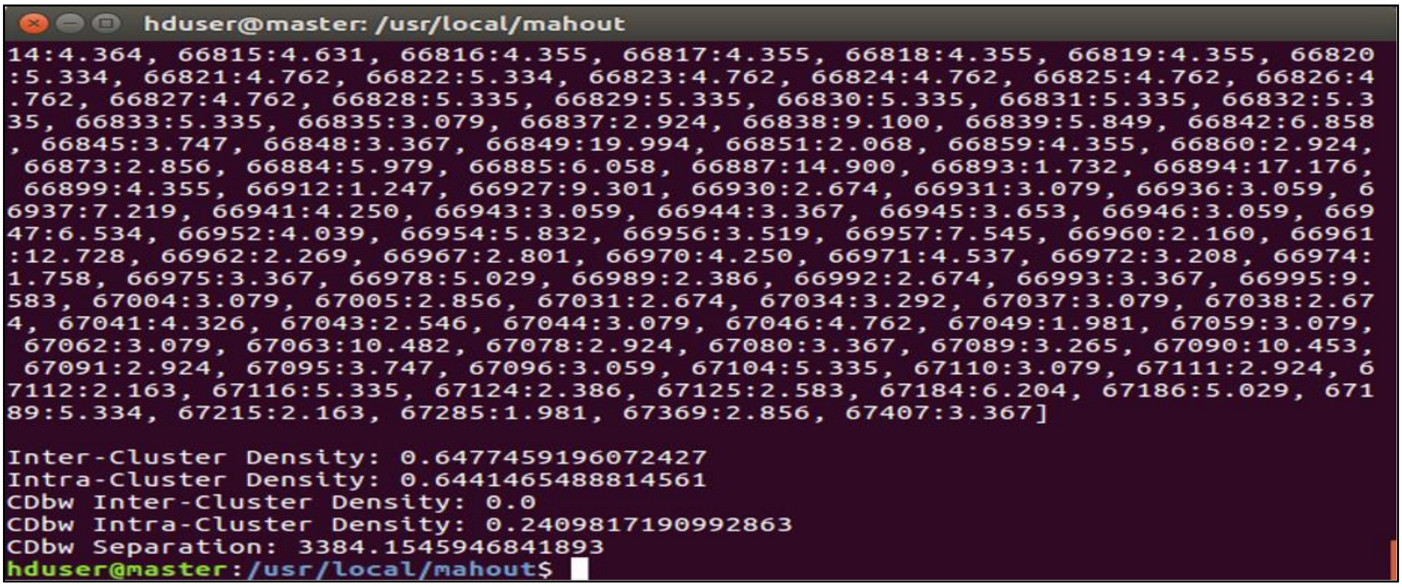


```
 66816:4.355, 66817:4.355, 66818:4.355, 66819:3.079, 66820:3.079, 66821:3.367, 6
6822:3.079, 66823:3.367, 66824:3.367, 66825:3.367, 66826:3.367, 66827:3.367, 668
34:3.375, 66835:3.079, 66837:2.924, 66838:8.408, 66839:5.471, 66840:3.079, 66842
:6.539, 66845:3.059, 66848:3.367, 66849:14.418, 66850:3.367, 66851:2.924, 66855:
4.355, 66856:5.335, 66870:5.335, 66873:2.856, 66874:3.079, 66878:2.674, 66880:8.
909, 66882:3.367, 66885:4.833, 66887:15.684, 66893:1.732, 66894:17.088, 66896:3.
208, 66898:4.762, 66905:3.367, 66907:4.039, 66908:2.520, 66910:1.521, 66927:7.45
7, 66936:3.059, 66937:7.718, 66941:3.292, 66943:4.326, 66945:3.164, 66946:4.837,
 66948:2.674, 66954:3.367, 66956:2.975, 66960:1.247, 66961:10.649, 66964:2.700,
66967:3.431, 66970:2.688, 66971:6.002, 66972:6.806, 66974:3.515, 66977:3.802, 66
978:3.802, 66989:2.386, 66992:2.674, 66995:8.629, 66997:2.674, 67000:3.367, 6701
1:2.856, 67012:5.335, 67020:2.674, 67031:2.674, 67034:2.688, 67040:3.367, 67041:
2.163, 67043:2.940, 67045:2.386, 67046:3.367, 67054:2.674, 67055:2.674, 67058:2.
856, 67063:10.293, 67067:3.079, 67077:2.674, 67078:2.924, 67080:3.367, 67086:2.8
56, 67089:4.898, 67090:10.195, 67096:3.747, 67102:2.674, 67111:2.924, 67184:3.58
2, 67186:4.656, 67189:3.079, 67215:2.163, 67219:5.335, 67369:2.856, 69567:3.059,
 69570:3.059]

Inter-Cluster Density: 0.391399986871574
Intra-Cluster Density: NaN
CDbw Inter-Cluster Density: 0.0
CDbw Intra-Cluster Density: 0.0
CDbw Separation: 552668.6420992917
hduser@master:/usr/local/mahout$ 
```

Fig 7 Fuzzy K Means Clustering Evaluator Using Canopy

Figure 8 shows the output of clusterdump for fuzzy k means clustering where 10 clusters are formed with inter cluster density of 0.64, intra cluster density of 0.64 and CDbw separation of 3384 approximately.



```
14:4.364, 66815:4.631, 66816:4.355, 66817:4.355, 66818:4.355, 66819:4.355, 66820
:5.334, 66821:4.762, 66822:5.334, 66823:4.762, 66824:4.762, 66825:4.762, 66826:4
.762, 66827:4.762, 66828:5.335, 66829:5.335, 66830:5.335, 66831:5.335, 66832:5.3
35, 66833:5.335, 66835:3.079, 66837:2.924, 66838:9.100, 66839:5.849, 66842:6.858
, 66845:3.747, 66848:3.367, 66849:19.994, 66851:2.068, 66859:4.355, 66860:2.924,
 66873:2.856, 66884:5.979, 66885:6.058, 66887:14.900, 66893:1.732, 66894:17.176,
 66899:4.355, 66912:1.247, 66927:9.301, 66930:2.674, 66931:3.079, 66936:3.059, 6
6937:7.219, 66941:4.250, 66943:3.059, 66944:3.367, 66945:3.653, 66946:3.059, 669
47:6.534, 66952:4.039, 66954:5.832, 66956:3.519, 66957:7.545, 66960:2.160, 66961
:12.728, 66962:2.269, 66967:2.801, 66970:4.250, 66971:4.537, 66972:3.208, 66974:
1.758, 66975:3.367, 66978:5.029, 66989:2.386, 66992:2.674, 66993:3.367, 66995:9.
583, 67004:3.079, 67005:2.856, 67031:2.674, 67034:3.292, 67037:3.079, 67038:2.67
4, 67041:4.326, 67043:2.546, 67044:3.079, 67046:4.762, 67049:1.981, 67059:3.079,
 67062:3.079, 67063:10.482, 67078:2.924, 67080:3.367, 67089:3.265, 67090:10.453,
 67091:2.924, 67095:3.747, 67096:3.059, 67104:5.335, 67110:3.079, 67111:2.924, 6
7112:2.163, 67116:5.335, 67124:2.386, 67125:2.583, 67184:6.204, 67186:5.029, 671
89:5.334, 67215:2.163, 67285:1.981, 67369:2.856, 67407:3.367]

Inter-Cluster Density: 0.6477459196072427
Intra-Cluster Density: 0.6441465488814561
CDbw Inter-Cluster Density: 0.0
CDbw Intra-Cluster Density: 0.2409817190992863
CDbw Separation: 3384.1545946841893
hduser@master:/usr/local/mahout$ 
```

Fig 8 Fuzzy K-Means Clustering with 10 Clusters

The utility clusterdump of mahout use certain arguments in order to show the output of cluster in desired format [22]. -i is input (path to job input directory), -o is output directory pathname for output which is the path on local file system instead of HDFS, -p is the directory containing points sequence files that map input vectors to their cluster which output the points associated with the cluster, and *n* is the number of top terms to print –of output the file in any of three formats i.e. text, CSV or graphml file.The output file generated consists of *n* which is the number of elements in the cluster identifier *r=[z, ...]*: is the radius of the cluster and the identifier signifying the clusters; *c=[z, ...]*:is the centroid of the cluster, with the z's being the weights of the different dimensions.

Table 1 below shows the result of all three clustering algorithms and it can be clearly observed that feeding K-Means clustering and fuzzy K-Means clustering with canopy generated clusters is not showing good results as compared to the results with lesser number of clusters.

Table 1 Clustering Evaluation Results

Parameter	**Canopy**	**K-Means**		**Fuzzy K-Means**	
		K-I	**K-II**	**FK-I**	**FK-II**
No. Of Clusters	32	32	10	32	10
Inter-Cluster Density	-	-	0.54	0.31	0.64
Intra-Cluster Density	0	0	0.59	-	0.64
CDbw Inter-Cluster Density	0	0	0	0	0
CDbw Intra-Cluster Density	-	-	0.43	0	0.24
CDbw Separation	0	0	14435	552668	3384

IV. CONCLUSION AND FUTURE WORK

Healthcare industry is gathering data from all over the world in all possible forms. To store and analyze that large amount of data is big task if used tradition methods. But big data analytics comes here to rescue. Big data analysis if used in healthcare can take it to whole other better level. *Various techniques are used for scaling Big Data to make it comprehensible and intelligible. Clustering is one of them.* It is unsupervised learning which basically works on collection of data that is unlabelled and then analysed to identify a pattern, so that the data can be grouped together. Thus, clustering discovers groups of similar features together.

This paper deals with the real-world data and collects annual report generated by various hospitals in India for last ten years. It has been observed that data mining techniques using tools like Weka are not able to process big data so Hadoop is used instead. Hadoop provides a machine learning library to process big data efficiently. Three clustering techniques namely canopy clustering, K-Means clustering and fuzzy means clustering are used for evaluating the healthcare data. It has also been observed that k-means and fuzzy clustering are not generating good output when passed with canopy as initial cluster seed whereas performs better if lesser clusters are formed.

For future work the same real-world dataset can be treated with other machine learning algorithms to get better insights from the data.

➢ *Conflicts of Interest*

All authors have no conflicts of interest to declare.

REFERENCES

[1]. Prachi Surwade, Prof. Satish S. Banait, "A Survey on Clustering Techniques for Mining Big Data", International Journal of Advanced Research in Science and Management, Feburary 2016, 2(2)

[2]. Apache Mahout: https://mahout.apache.org/

[3]. T. Sajana, C. M. Sheela Rani and K. V. Narayana, "A Survey on Clustering Techniques for Big Data Mining", Indian Journal of Science and Technology, January, 2016,9(3)

[4]. Miss. Harshada S. Deshmukh, Prof. P. L. Ramteke, "Comparing the Techniques of Cluster Analysis for Big Data", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET, December 2015, 4(12)

[5]. Keshav Sanse, Meena Sharma, "Clustering methods for Big data analysis", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), March 2015, 4(3)

[6]. Dr. Venkateswara Reddy Eluri, MS. Amina Salim Mohd AL-Jabri, Dr. M. Ramesh, Dr. Mare Jane, "A Comparative Study of Various Clustering Techniques on Big Data Sets using Apache Mahout", 3rd MEC International Conference on Big Data and Smart City,2016

[7]. Fatos Xhafa, Adriana Bogza, Santi Caballé, "Performance Evaluation of Mahout Clustering Algorithms Using a Twitter Streaming Dataset" IEEE 31st International Conference on Advanced Information Networking and Applications, 2017

[8]. Van-Dai Ta, Chuan-Ming Liu, Goodwill Wandile Nkabinde, "Big Data Stream Computing in Healthcare Real-Time Analytics" IEEE International Conference on Cloud Computing and Big Data Analysis, 2016

[9]. Rui Máximo Esteves, Chunming Rong, "Using Mahout for clustering Wikipedia's latest articles: A comparison between k-means and fuzzy c-means in the cloud" Third IEEE International Conference on Cloud Computing Technology and Science, 2011

[10]. Ahmad Al-Khoder, Hazar Harmouch, “Evaluating four of the most popular Open Source and Free Data Mining Tools” IJASR International Journal of Academic Scientific Research, 2015, 3(1)

[11]. Hoda A. Abdel Hafez, “Mining Big Data in Telecommunications Industry: Challenges, Techniques, and Revenue Opportunity” Dubai UAE Jan 28-29, 2016

[12]. Amini A, Wah TY, Saboohi H. On density-based data streams clustering algorithms: A survey. Journal of Computer Science and Technology, Jan. 2014,29(1):116-141

[13]. Olga Kurasova, Virginijus Marcinkevicius, Viktor Medvedev, Aurimas Rapecka, and Pavel Stefanovi, “Strategies for Big Data Clustering” IEEE 26th International Conference on Tools with Artificial Intelligence, 2014

[14]. Pritika Talwar, Shubham, Komalpreet Kaur, “Exploring Clustering techniques in Machine Learning”, International Journal of Creative Research Thoughts (IJCRT), March 2024,12(3)

[15]. Aasim Ayaz Wani, “Comprehensive analysis of clustering algorithms: exploring limitations and innovative solutions” Peer J Computer science, https://doi.org/10.7717/peerj-cs.2286, August 2024

[16]. Anju Parmar, Divya Chauhan, Dr. K.L. Bansal, “Performance Evaluation of Weka Clustering Algorithms on Large Datasets” International Journal of Advanced Research, 2017,5(6), 2209-2216

[17]. Annual Reports: https://www.aiims.edu/en/about-us/annual-reports.html: accessed on: 20th january, 2019

[18]. Tmc-Annual Report: https://tmc.gov.in/index.php/tmc-annual-report, Accessed on: 21st august, 2019

[19]. Fortis bmw reports: https://www.fortismalar.com/bmw-report, accessed on: 30th January, 2019

[20]. Apollo Hospitals: https://www.apollohospitals.com/corporate/investor-relations/financial-reports, accessed on 20th January, 2019

[21]. Linux Uprising: https://www.linuxuprising.com/2019/05/how-to-convert-pdf-to-text-on-linux-gui.html

[22]. Clustering your data: https://mahout.apache.org/users/clustering/clusteringyourdata.html