

**CSCI 5408**

**Summer 2018**

**PROJECT REPORT**

**Document Clustering Using K-Means Algorithm**

**Submitted by:-**

**Nikhil Dhirmalani: B00775542**

**Khushboo Siwal: B00781497**

**TABLE OF CONTENTS**

1. Objective
2. Value Proposition
3. Implementation
4. Data Set
5. Algorithms Used
6. Visualizations
7. Programming Language and Tools
8. Roles and Responsibilities
9. Work Breakdown Structure
10. Limitations
11. Future Work
12. Critical Review
13. Code Submission
14. References

**LIST OF TABLES**

[Table 1: List of 20 newsgroups 5](#_Toc521242693)

[Table 2: Phases of the project 12](#_Toc521242694)

**LIST OF FIGURES**

[Figure 1 Steps for Implementation 5](#_Toc521242616)

[Figure 2 K-Means Algorithm 7](#_Toc521242617)

[Figure 3 Clusters after K-means implementation 8](#_Toc521242618)

[Figure 4 Clusters after term matrix fed to K-means algorithm using sklearn library 8](#_Toc521242619)

[Figure 5 Clusters plotted in matplotlib 9](#_Toc521242620)

[Figure 6 Clusters obtained from K-means algorithm implemented using sklearn 10](#_Toc521242621)

[Figure 7 IRIS dataset results from our algorithm 11](#_Toc521242622)

**Objective**

Document clustering is the act of collecting similar documents into bins, where similarity is some function on a document. Manually categorizing and grouping text sources can be extremely laborious and time-consuming, especially for publishers, news sites, blogs or anyone who deals with a lot of content. The objective of this project is to perform document clustering in order to find similarities between document based on words in the document and cluster relevant documents together using K-means algorithm. The first step in the Clustering process is to create word vectors for the documents we wish to cluster. A vector is simply a numerical representation of the document, where each component of the vector refers to a word, and the value of that component indicates the presence or importance of that word in the document. The distance matrix between these vectors is then fed to algorithms, which group similar documents represented as vectors together into clusters.

**Value Proposition**

In this project, we will try to implement distributed Kmeans using map reduce in apache spark on our own. We will compare our K-Means implementation with the Kmeans library available from sklearn. Our primary focus is to learn about K-Means and perform clustering on documents in order to group relevant documents together.

Clustering Application:-

* Duplicate Content Detection : In many applications there is a need to find duplicates or near-duplicates in a large number of documents. Clustering is employed for plagiarism detection, grouping of related news stories and to reorder search results rankings (to assure higher diversity among the topmost documents). Note that in such applications the description of clusters is rarely needed.
* Recommendation System : In this application a user is recommended articles based on the articles the user has already read. Clustering of the articles makes it possible in real time and improves the quality a lot.
* Search Optimization : Clustering helps a lot in improving the quality and efficiency of search engines as the user query can be first compared to the clusters instead of comparing it directly to the documents and the search results can also be arranged easily.

**Implementation**

At the beginning we have eleven thousand documents each representing either politics, sport etc. from 20 News Group dataset. These documents undergo refinement which is fed to the algorithm to obtain clusters containing documents from similar domains. A document usually consists of huge number of words, it is not always necessary that each word is of importance. Due to which, a document has high dimensionality has to be reduced. Hence processing is carried on a document to reduce this dimensionality and get rid of extra words and to obtain weight of each of the word to be used in the algorithm. Conversion of documents into vectors is carried out. The main use of tokenization is identifying the meaningful keywords called tokens. Tokenization splits sentences into individual tokens, typically words.

Collected documents contain some unnecessary words by which dimensionality of a document will be increased; we should remove those words to get proper result. Pronoun, adverb, preposition etc. which are used constantly throughout in a document has to be removed. A Document-term matrix had to be formed where each row represents a document and each column are the total words in the dictionary. Each word is assigned TF-IDF score. For formation of this matrix we employ the idea from Vector Space Model (VSM)3 which is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers. This matrix of vectors is fed to the K-Means algorithm to find the clusters of similar documents, as these vectors have very high dimension we use PCA to reduce it to 3 dimensions and plot it for visualizing the clusters using Matplotlib and Plotly libraries.

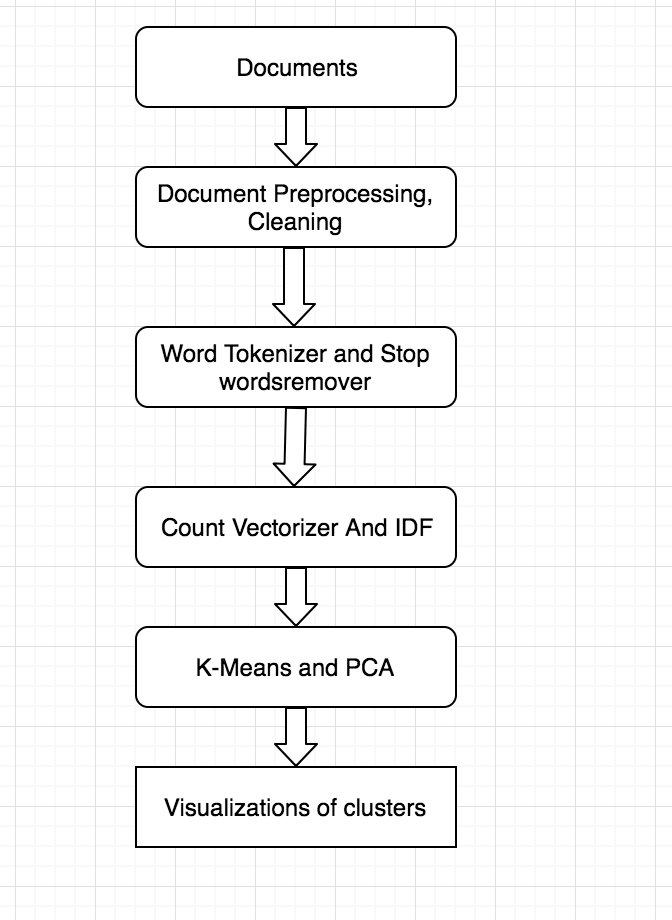


Figure 1 Steps for Implementation

**Data Set**

The **20 Newsgroups** data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups each corresponding to different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale / soc.religion.christian). Below table is the list of 20 newsgroups, partitioned (more or less) according to subject matter:

Table 1: List of 20 newsgroups

|  |  |  |
| --- | --- | --- |
| comp.graphics  comp.os.ms-windows.misc  comp.sys.ibm.pc.hardware  comp.sys.mac.hardware  comp.windows.x | rec.autos  rec.motorcycles  rec.sport.baseball  rec.sport.hockey | sci.crypt  sci.electronics  sci.med  sci.space |
| misc.forsale | talk.politics.misc  talk.politics.guns  talk.politics.mideast | talk.religion.misc  alt.atheism  soc.religion.christian |

Its details are as follows:

• Number of unique documents = 18,828

• Number of categories = 20

• Number of unique words after removing the stop words = 71,830

The IRIS data sets consists of 3 different types of irises’ (Setosa, Versicolour, and Virginica) petal and sepal length, stored in a 150x4 numpy.ndarray.

The rows being the samples and the columns being: Sepal Length, Sepal Width, Petal Length and Petal Width.

**Algorithms Used**

Clustering algorithms groups documents into groups called clusters, where the algorithm's goal is to create internally coherent clusters that are distinct from one another. We used k-means algorithm in our project to perform document clustering.

k-means clustering is a method of [vector quantization](https://en.wikipedia.org/wiki/Vector_quantization), originally from [signal processing](https://en.wikipedia.org/wiki/Signal_processing), that is popular for [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) in [data mining](https://en.wikipedia.org/wiki/Data_mining). k-means clustering aims to [partition](https://en.wikipedia.org/wiki/Partition_of_a_set) n observations into k clusters in which each observation belongs to the [cluster](https://en.wikipedia.org/wiki/Cluster_(statistics)) with the nearest [mean](https://en.wikipedia.org/wiki/Mean), serving as a prototype of the cluster.  This results in a partitioning of the data space into [Voronoi cells](https://en.wikipedia.org/wiki/Voronoi_cell).

**K-MEANS Algorithm**

In the clustering problem, we are given a training set x(1),...,x(m)x(1),...,x(m), and want to group the data into a few cohesive "clusters." Here, we are given feature vectors for each data point \ x(i)∈Rn as usual; but no labels y(i) (making this an unsupervised learning problem). Our goal is to predict k centroids **and** a label c(i) for each datapoint. The k-means clustering algorithm is as follows:

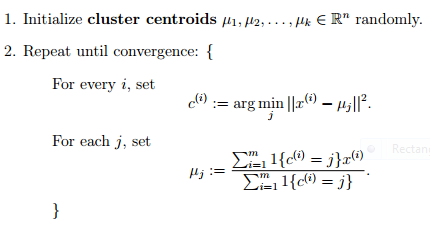


Figure 2 K-Means Algorithm

**TF-IDF Algorithm (Document Term Matrix Score)**

TF\*IDF is an information retrieval technique that weighs a term’s frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF\*IDF weight of that term.

**Put simply, the higher the TF\*IDF score (weight), the rarer the term and vice versa.**

The TF\*IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the web, which is referred to as *corpus*.

For a term **t** in a document **d**, the weight **Wt,d** of term t in document d is given by:

Wt,d = TFt,d log (N/DFt)

Where:

* TFt,d is the number of occurrences of t in document d.
* DFt is the number of documents containing the term t.
* N is the total number of documents in the corpus.

**Visualizations**

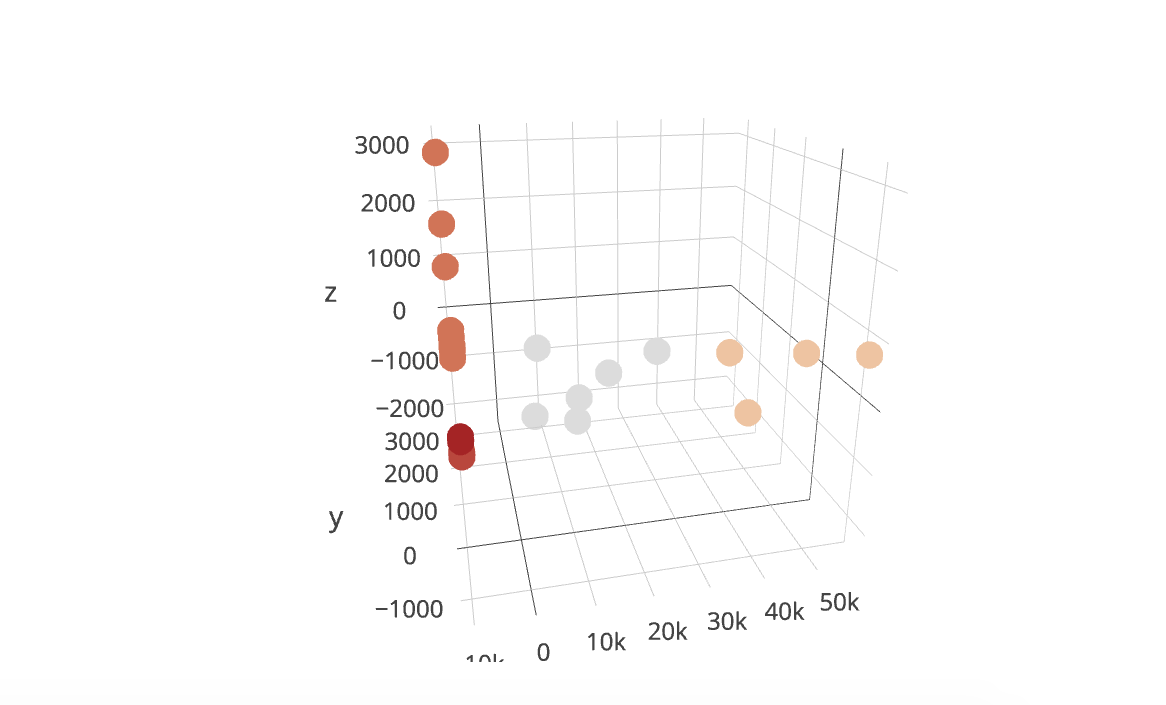


Figure 3 Clusters after K-means implementation

As shown in above figure the clusters that we have got when we have used our implemented K-Means algorithm. Here grey clusters represent documents that were based on religion. Orange color cluster depicts documents are related to sports. As we ran our algorithm for small epochs we were not getting the accurate results.

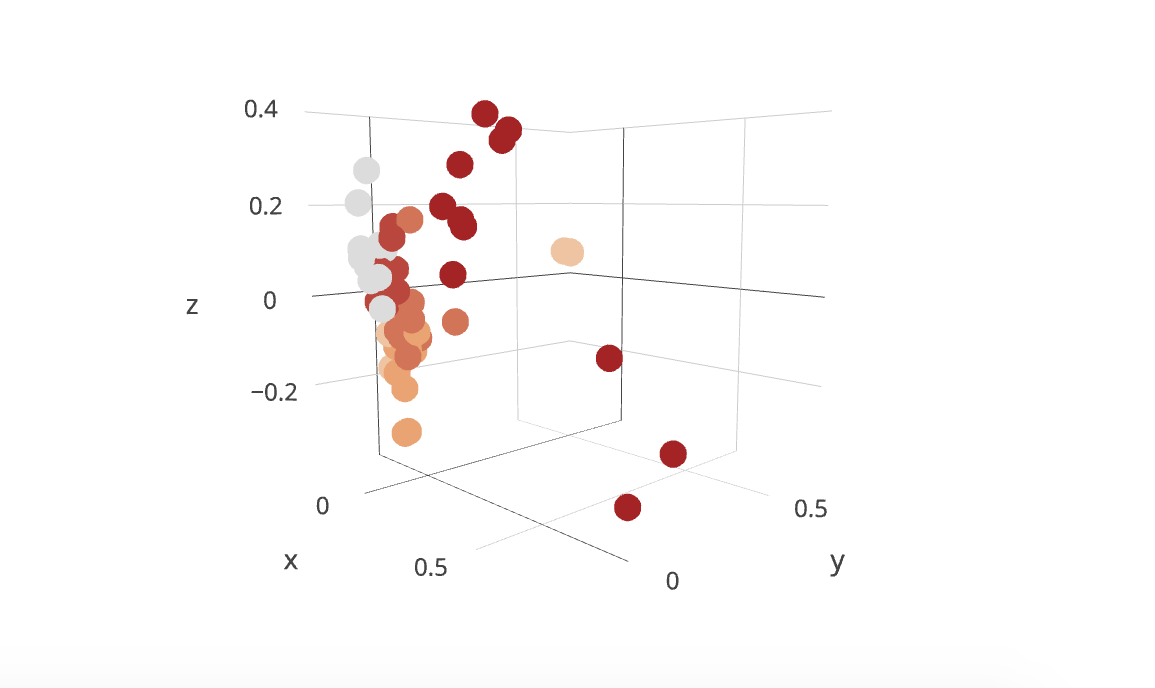


Figure 4 Clusters after term matrix fed to K-means algorithm using sklearn library

As shown in above figure these are the clusters that we get on feeding our document term matrix to K-Means algorithm implemented using sklearn library. We can see that we get all the related documents clustered and we get more clusters as we run the algorithm for 300 epochs. Here the orange clusters represent documents on religion and maroon clusters represent documents on sports.

The above figures are plotted using plotly, we have given our results on 6 clusters. We can run our algorithm on different clusters and get different visualizations.

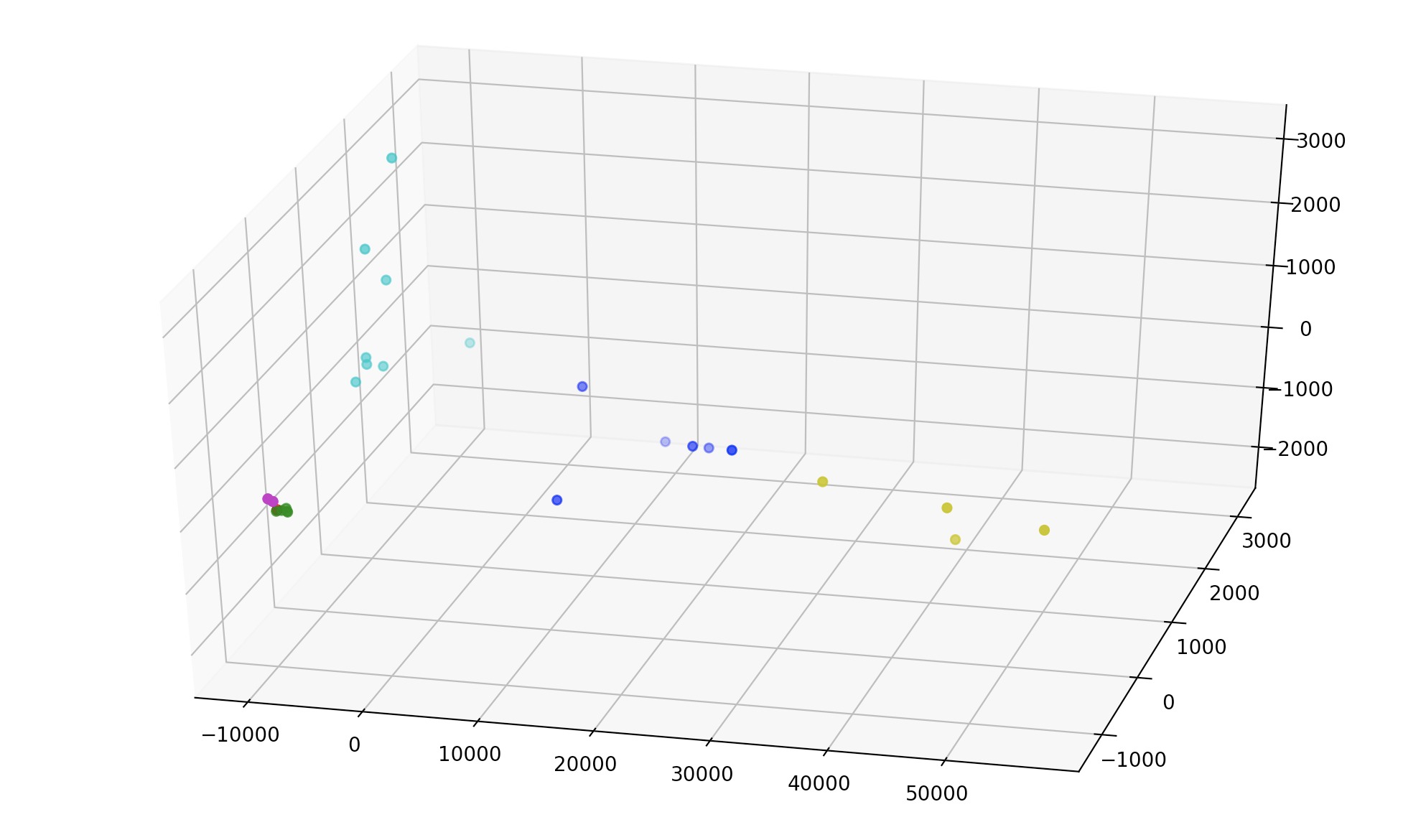


Figure 5 Clusters plotted in matplotlib

As shown above we plotted the clusters in matplolib, the light blue clusters represent documents on religion and dark blue represents documents on sports.

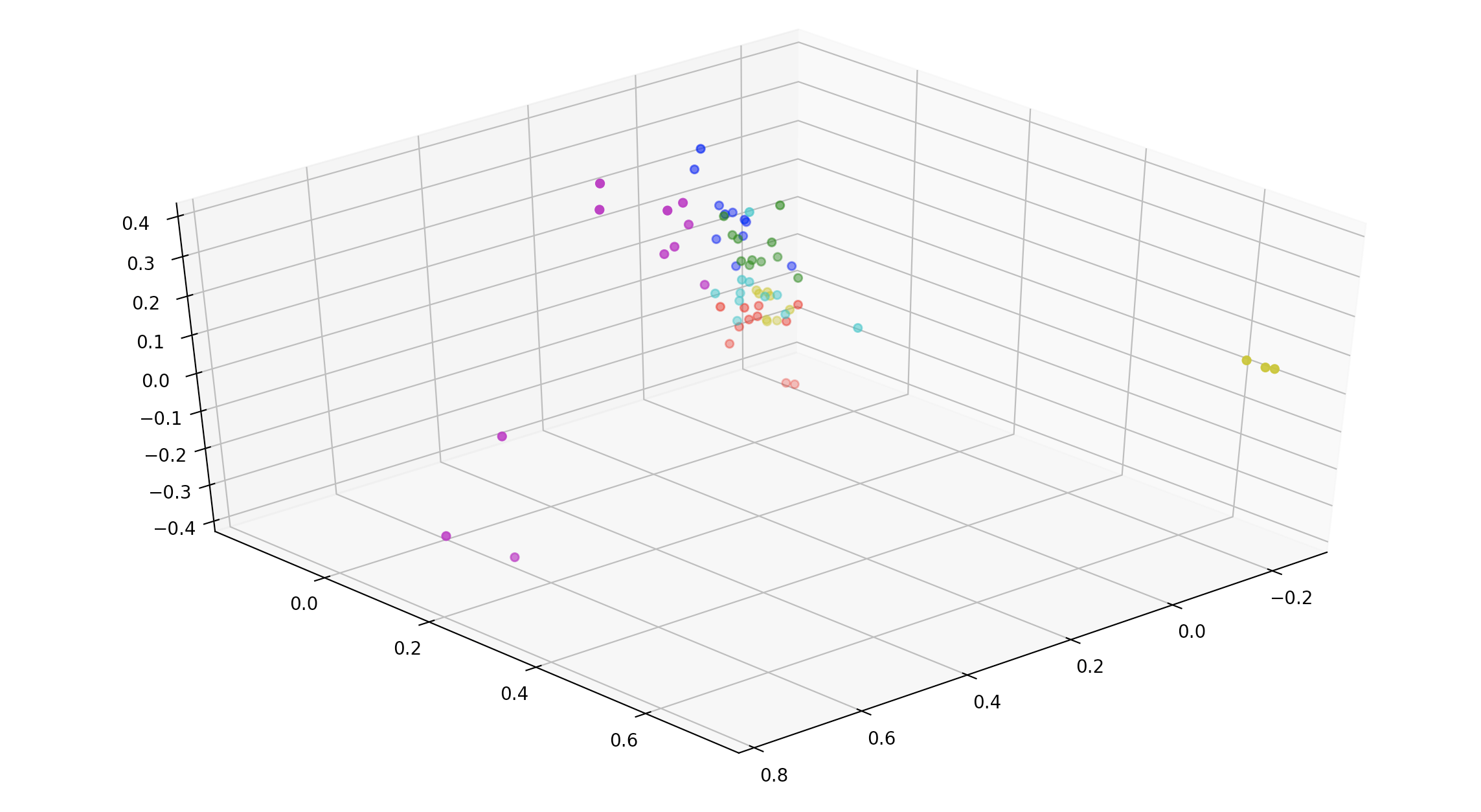


Figure 6 Clusters obtained from K-means algorithm implemented using sklearn

The above diagram is plotted for clusters obtained from K-Means algorithm implemented using the sklearn. We can see that we get all six clusters equally in opposite to our algorithm which on running for more epochs can give good results.

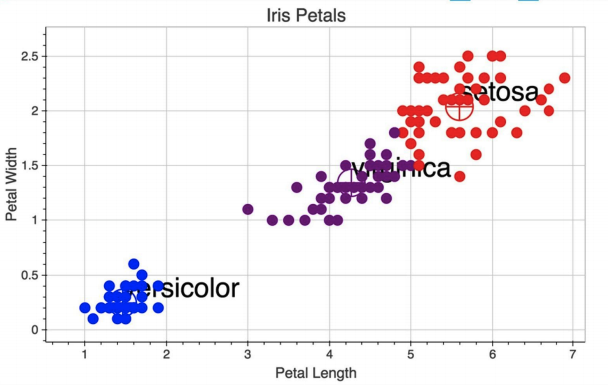


Figure 7 IRIS dataset results from our algorithm

In order to check the veracity of our algorithm we have given our results on IRIS dataset. We can clearly see we get all the clusters each representing the type of flower. This proves that the algorithm that we implemented is correct, but it takes time to run on big datasets.

**Programming Language and Tools**

We used following tools and languages for extraction, transformation and loading process:

1. Python 2.7 as the programming language
2. Pycharm IDE to run the python files
3. Numpy, Pandas python packages
4. Matplotlib and Plotly for data visualization
5. Pyspark framework for data processing
6. Sklearn for PCA, K-Means, IRIS and 20 News Group datasets.

**Roles and Responsibilities**

* Nikhil Dhirmalani(B00775542): Nikhil worked with Khushboo in preparing project proposal, tracking sprint progress, presentation slides and project report. Nikhil set up the environment for apache spark and implement distributed Kmeans using map-reduce and used this model in order to cluster the data. Nikhil also implemented sklearn K-Means to verify and compare the results of implemented algorithm.
* Khushboo Siwal(B00781497): Khushboo worked with Nikhil in preparing project proposal, tracking sprint progress, presentation slides and project report. Khushboo set up the environment for Plotly, Matplolib and she preprocessed the data, created document term matrix and visualized the results from trained model using different graphs.

**Work Breakdown Structure**

We have divided the project into five phases and each phase corresponds to a sprint. We have then added each task that we have to perform in each phase in the table given below.

Table 2: Phases of the project

|  |  |  |
| --- | --- | --- |
| **Level1** | **Level2** | **Level3** |
| 1 Document Clustering using Kmeans Algorithm  1 Document Clustering using Kmeans Algorithm | 1.1 Initiation | 1.1.1 Determine Project Team  1.1.2 Project Team Kickoff Meeting  1.1.3 Discussion on project, dataset, algorithm, tools and programming language.  1.1.4 Review on various works used to solve the problem  1.1.5 Project work division and sprints development  1.1.6 Review each team members work and suggest changes if required  1.1.7 Develop Project Proposal and Submit.  1.1.8 Milestone: Project Proposal Approval |
| 1.2 Planning | 1.2.1 Project Team Meeting  1.2.2 Discussion on work to be accomplished in ongoing sprint  1.2.3 Download softwares and tools used in project and environment setup on machines.  1.2.4 Data preprocessing tasks and analysis of algorithm used in project.  1.2.5 Review each team members work and suggest changes if required |
| 1.3 Execution | 1.3.1 Project Team Meeting  1.3.2 Discussion on work to be accomplished in ongoing sprint  1.3.3 Implement the algorithm and train a model on preprocessed dataset  1.3.4 Discussion on results and methods to be implemented to improve the obtained results |
| 1.4 Experiment and Visualization | 1.4.1 Project Team Meeting  1.4.2 Discussion on work to be accomplished in ongoing sprint  1.4.3 Experiment with model and try to find different results.  1.4.4 Use visualization tool in order to show the results |
| 1.5 Closeout | 1.5.1 Discuss experiments performed,results and findings  1.5.2 Presentation slides preparation and review  1.5.3 Develop Project Report and Submit  1.5.4 Project Sponsor Reviews Project Report  1.5.5 Project Report Signed/Approved |

**Limitations**

We have encountered many limitations in our approach, when selecting the initial centroids, we are choosing randomly which has chances that two centroids selected must be close to each other. We want the initially selected centroids as far from each other. Moreover, our algorithm is also less efficient it takes almost 20s to run a single epoch whereas the K-Means implemented through sklearn takes only 20s to run 300 epochs. We have also not implemented a method or algorithm to evaluate the clusters that we get from our algorithm. (Ex Shanon’s Entropy and F-measure).

We have used Euclidian distance as similarity measure instead we can use Cosine Similarity or Averaged Kullback-Leibler Divergence.

**Future Work**

Our approach is very simple we are just creating clusters and algorithm which we have implemented is also very simple (i.e. K-Means). We would like to explore more novel approaches like LDA Topic modeling for the document clustering. We will like to follow the approach from the given paper and also like to create an interface for document exploration as shown in below link.

LDA based Topic models

A.J.B. Chaney and D.M. Blei, “Visualizing Topic Models,” Proc. Sixth Int’l AAAI Conf. Weblogs and Social Media (AAAI ICWSM), pp. 419-422, 2012.

<http://www.princeton.edu/~achaney/tmve/wiki100k/browse/topic-presence.html>

**Challenges in Document Clustering**

1. Selection of appropriate features of the documents.
2. Selection of appropriate similarity measure
3. Selection of appropriate clustering method
4. Assessment of the quality of the clusters.
5. Implementation of the clustering algorithm in an efficient way by making optimal use of available memory and CPU resources.

**Critical Review**

Document Clustering is a technique used to group documents into groups(clusters) based on similarity of content. The organization is to make documents more understandable and easier to search the relevant information, easier to process, even more efficient in utilizing communication bandwidth and storage space. An example is clustering results of a web search engine operation into groups of similar documents. The choice of algorithm is a big issue for the developers. K-means is the most popular clustering algorithm. Just like others, it also has its own weaknesses.

**Code Submission**

<https://github.com/NK273610/DataWarehouseAssignmentProject>

**References**

### ANON

**In-text:**(2018)

**Your Bibliography:**(2018). Retrieved from https://www.linkedin.com/pulse/nlp-text-analytics-simplified-document-clustering-parsa-ghaffari/

### ANON

In-text: (2018)

Your Bibliography: (2018). Retrieved from https://www.linkedin.com/pulse/20141209180635-83626359-nlp-and-text-analytics-similified-document-classification/

### HOME PAGE FOR 20 NEWSGROUPS DATA SET

**In-text:**("Home Page for 20 Newsgroups Data Set", 2018)

**Your Bibliography:**Home Page for 20 Newsgroups Data Set. (2018). Retrieved from <http://qwone.com/~jason/20Newsgroups/>

### DOCUMENT CLUSTERING

**In-text:**("Document clustering", 2018)

**Your Bibliography:**Document clustering. (2018). Retrieved from https://en.wikipedia.org/wiki/Document\_clustering

1. Document Clustering using Improved K-means Algorithm Anjali Vashist1 , Rajender Nath
2. Document Clustering using K-Means and K-Medoids Rakesh Chandra Balabantaray\*, Chandrali Sarma\*\*, Monica Jha\*\*\*
3. (2018). Retrieved from <http://shodhganga.inflibnet.ac.in/bitstream/10603/140299/14/14_conclusion%20and%20scope%20for%20future%20research.pdf>
4. <https://pdfs.semanticscholar.org/8970/5ababa239d63c02d27511cffd3ab7d30f44c.pdf>
5. <https://www.elephate.com/blog/what-is-tf-idf/>
6. <http://stanford.edu/~cpiech/cs221/handouts/kmeans.html>