

## HOMEWORK 01

### I. DATA : 20NewsGroup

- It's a data set that has twenty distinct newsgroups, which corresponds to topics of different categories like science, computer, sports, religion, politics, miscellaneous.
- Few news groups are related to each other and hence can fall into one of the 6 categories mentioned above
- Each newsgroup has approximately 1k documents and each document consists of words that are in context of the topic under which it is.
- The documents in comp.graphics and comp.sys.ibm.pc.hardware are somehow related to each other
- But the documents in talk.politics.guns and comp.graphics are not related to one another
- The # of news topics are 6 and the no. of documents per topic and the no. of unique terms per topic are as follows:

| 20NewsGroups             | # of Documents | # of Documents per group | # of Unique Words | # of Unique Words per group |
|--------------------------|----------------|--------------------------|-------------------|-----------------------------|
| comp.graphics            | 584            | 2936                     | 13071             | 86826                       |
| comp.os.ms-windows.misc  | 591            |                          | 39038             |                             |
| comp.sys.ibm.pc.hardware | 590            |                          | 11106             |                             |
| comp.sys.mac.hardware    | 578            |                          | 10144             |                             |
| comp.windows.x           | 593            |                          | 13467             |                             |
| misc.forsale             | 585            | 585                      | 11392             | 11392                       |
| rec.autos                | 594            | 2389                     | 11661             | 46061                       |
| rec.motorcycles          | 598            |                          | 11871             |                             |
| rec.sport.baseball       | 597            |                          | 10222             |                             |
| rec.sport.hockey         | 600            |                          | 12307             |                             |
| sci.crypt                | 595            | 2373                     | 15260             | 58043                       |
| sci.electronics          | 591            |                          | 11648             |                             |
| sci.med                  | 594            |                          | 15768             |                             |
| sci.space                | 593            |                          | 15367             |                             |
| alt.atheism              | 480            | 3031                     | 11564             | 81837                       |
| soc.religion.christian   | 599            |                          | 13365             |                             |
| talk.politics.guns       | 546            |                          | 15799             |                             |
| talk.politics.mideast    | 564            |                          | 15987             |                             |
| talk.politics.misc       | 465            |                          | 13607             |                             |
| talk.religion.misc       | 377            |                          | 11515             |                             |

### I. DATA-YELP:

- It is a dataset that has reviews for various local businesses and the dataset is in JSON format and has 5 files in it. It is of 5.39 GB.
- The data here is very much diverse, vast and requires lot of cleaning to reduce processing
- The JSON format of the dataset is to be converted to csv file. The csv file has business ID and text in it.
- There are 5000 rows and 18240 terms in the csv file

### II. EXPERIMENT

#### II. A. DATA PREPROCESSING: NEWSGROUPS

#### ANALYSIS ON GROUPS PICKED

- The groups that I selected are following and which was of size 9.94MB: (I) comp.graphics (II) comp.sys.ibm.pc.hardware, (III) talk.politics.guns, (IV) talk.politics.mideast, (V) talk.religion.misc
- From the selected groups comp.graphics and comp.sys.ibm.pc.hardware are related to each other in regards of computer. Also talk.politics.guns,talk.politics.mideast are related in regards with “politics” and talk.religion.misc is not related to any which seems to be in regards to “religion”.
- If the terms in the groups are related then they will fall under same cluster else in different cluster
- It should form 3 clusters by kmeans
- LSA,LDA will give better results as compared to kmeans as the document is related to the topics which are based on the words that we get using dtm

### CREATE DOCUMENT TERM MATRIX

- Dtm is a matrix in which each row is a document and each column is the term.
- The dtm of the 5 groups that I selected has 2661 documents and 41962 terms

### REMOVE STOP WORDS

- As the stop words such as this,that are very frequently occurring in the document but they don't have carry any meaningful information hence we need to remove them so that the processing time to process those words will reduce and also we will get better and precise results.
- They are removed using tm\_map function in tm package
- After removing the stops words the terms in the dtm is reduced from 41962 to 30008
- There was a reduction of 28.49% in the terms

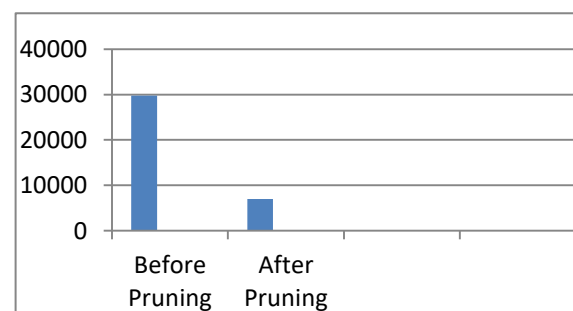
### USE STEMMING:

- Here the derived words such as “segmentation” are reduced to its own root word “segment”
- After the stemming performed the dtm resulted in again reduction of terms from 30008 to 29753
- Hence there was 0.85% reduction in the no. of terms or 255 terms were reduced which is not too much

### PRUNE WORDS BY FREQUENCY:

- Here the words occurring in less than 4 doc or more than 350 doc. are removed.
- From the fig. (Right),we observe there is reduction in terms by 76.38% ie. from 29753 to 7026 hence there is tremendous decrease in the terms
- As the terms are reduced a lot, it will be helpful to obtain better output for LSA and LDA,as now the terms which are left over has a very meaningful information and has precise meaning which will also lead to better clustering.

**Fig. : Graph shows no. of terms before & after pruning**



- After the matrix is pruned and cleaned, the wordcloud was formed which showed the most frequent terms
- Most 10 frequent terms I found using R are : Armenian,drive,file,Israel,imag,isra,card,scsi,Turkish,support

**Fig.: Wordcloud for 5NewsGroups**



#### USE TF-IDF:

- It shows how important or how significant the word is.
- The full form of tf-idf is term frequency inverse document frequency
- The tf-idf of the pruned dtm resulted in the no. of documents as 2661 and the no. of terms as 7026

**Fig.: tf-idf weights**

|       | abstract   | accept     | address    | affili     | applic     | april      | aspect     | attend     | author     | avail      | c |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|---|
| 37261 | 0.09286994 | 0.04812535 | 0.07204629 | 0.04289106 | 0.03304413 | 0.02824993 | 0.03652163 | 0.08968312 | 0.07394824 | 0.02369135 |   |
| 37913 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37914 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.06762802 |   |
| 37915 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.10375856 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37916 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37917 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37918 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37921 | 0.00000000 | 0.07407529 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.07293218 |   |
| 37922 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37923 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |   |
| 37924 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.08265647 |   |
| 37925 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.12093619 | 0.00000000 |   |

Showing 1 to 14 of 2,661 entries

## II.A. DATA PREPROCESSING : YELP

- There is a need of processing the yelp data before we start clustering it as the data is vast and hence if we extract only the meaningful information then the processing will be easy and will get better clustering results for kmeans,LSA,LDA

### CREATE DOCUMENT TERM MATRIX

- The document term matrix of yelp dataset was made and I got 5000 rows and 18240 terms
- Below is how the csv file was read in R,read.csv() is used to read file and as the corpus has csv file in it so VectorSource() is used and the encoding used is UTF-8

```
> yelp<-c("/home/ubuntu/Yelp.csv")
> yelp_read = read.csv(yelp)
> yelp_corpus <- Corpus(VectorSource(iconv(yelp_read$text, 'UTF-8', 'ASCII')))
> yelp_corpus1=yelp_corpus
> yelp_dtm<-DocumentTermMatrix(yelp_corpus1)
> yelp_matrix<-as.matrix(yelp_dtm)
> dim(yelp_matrix)
[1] 5000 18240
```

## REMOVE STOP WORDS

- As we know stop words does not provide any useful information we need to remove them
- After removing it the terms are reduced from 18240 to 13496 ie. reduction of terms by 26.01%

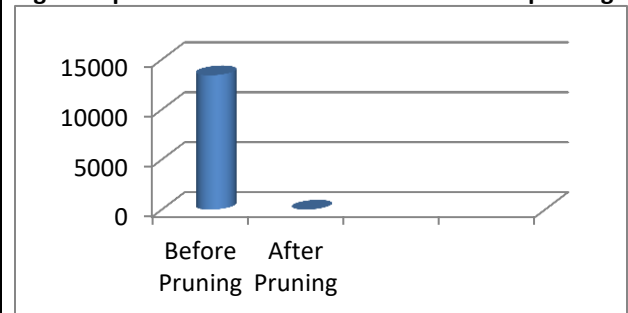
## USE STEMMING:

- After stemming the, the terms are reduced from 13496 to 13427, which results in reduction of terms by 0.51% i.e. by 69 terms

## PRUNE WORDS BY FREQUENCY:

- The terms occurring in more <4 doc and >350 doc are pruned here
- From the fig. (Right), the terms are reduced from 13427 to 3845 ie. 71.36% reduction
- The no. of terms are heavily reduced hence the clustering will give more better results and also LSA and LDA would form better clusters

Fig. : Graph shows no. of terms before & after pruning



- The wordcloud of the yelp dataset is shown on right
- Most 10 frequent terms for yelp that I found using R are : pizza,told,sauc,chee,flavor, home, room,server,lunch,check

```
> yelp_term_frequency_dec<-sort(freq,decreasing=TRUE)
> yelp_most_frequent_terms<-data.frame(names(yelp_term_frequency_dec),yelp_term_frequency_dec)
> head(yelp_most_frequent_terms,10)
      names.yelp_term_frequency_dec.  yelp_term_frequency_dec
pizza                             pizza                    562
told                              told                     500
sauc                              sauc                     442
chee                              chee                     440
flavor                            flavor                    440
home                              home                     432
room                              room                     426
server                            server                    425
lunch                             lunch                    416
check                             check                    408
```

Fig.: Wordcloud for yelp



## USE TF-IDF: Tf-idf of yelp is produced using weightTfidf()

```
yelp_dtm_pruned_tfidf<-weightTfidf(yelp_dtm_pruned)
yelp_matrix_pruned_tfidf<-as.matrix(yelp_dtm_pruned_tfidf)
view(yelp_matrix_pruned_tfidf)
```

**Analysis for both the data sets:** There was reduction in no. of terms at every data processing step for both the data sets.

## II. B. CLUSTERING EXPERIMENTS - NEWSGROUPS

### 1. CLUSTERING:

- Here the clustering was done using kmeans and with different index
- To get the no. of clusters NbClust was used.
- I used indexes such as duda,ball,silhouette all of them resulted in best no. of clusters as k=3
- Hence, I decided in taking k=3

Index=duda:

```
> no_clusters$Best.nc
Number_clusters Value_Index
3.0000         3.3276
> clus<-kmeans(matrix_pruned_tfidf,3)
> print(paste("SSE Value for matrix_pruned_tfidf :",clus$totss))
[1] "SSE Value for matrix_pruned_tfidf : 3243.94359954324"
```

Index=ball:

```
> no_clusters_ball <- NbClust(matrix_pruned_tfidf, min.nc=2, max.nc=15, method="kmeans", index="ball")
> no_clusters_ball$Best.nc
Number_clusters Value_Index
3.0000         542.3137
```

Index= silhouette

```
no_clusters_silhouette <- NbClust(matrix_pruned_tfidf, min.nc=2, max.nc=15, method="kmeans", index="silhouette")
no_clusters_silhouette$Best.nc
```

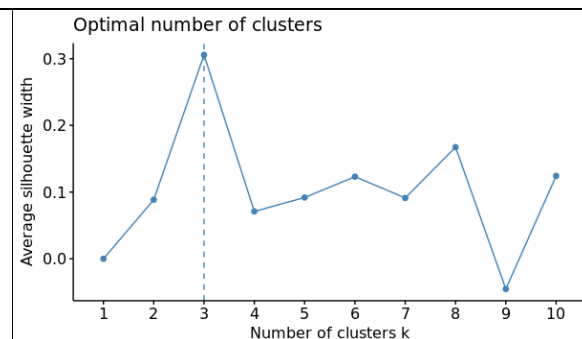


Fig.: Shows the optimal number of clusters i.e.K=3

```
clus<-kmeans(matrix_pruned_tfidf,3)
fviz_cluster(clus, data = matrix_pruned_tfidf,
geom = "point", stand = FALSE, frame.type = "norm")
```

Fig.: Code used to plot cluster(Right) by kmeans using fviz\_cluster

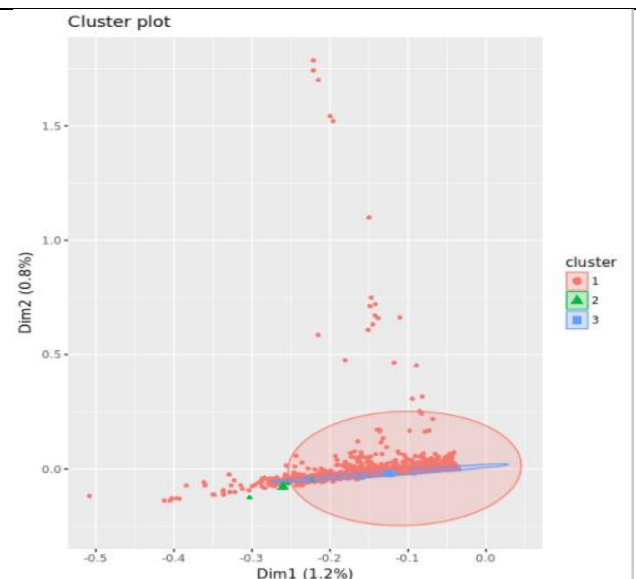


Fig.: Plotting the cluster for k=3 using kmeans

- From the figure above shows the cluster plotted using kmeans with 3 as best clusters

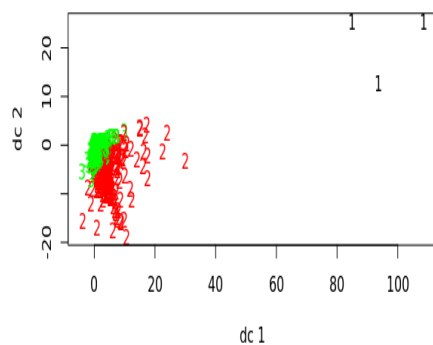
### 2. LSA-NEWSGROUPS:

- The SVD of the dtm was computed as : `svd_dec <- svd(dtm_pruned)`
- SVD helps in reducing the rank or the dimension of the matrix and the technique "LSA" uses SVD in order to get low dimensional matrix.

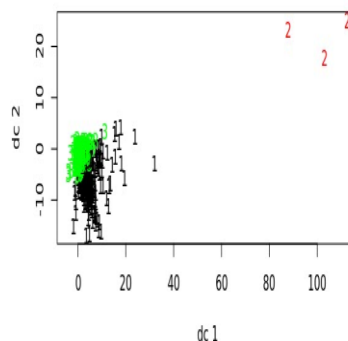
- The diagonal of the SVD is the singular value that is computed by and which maps the terms in latent semantic space
- **Computed D=50,100,200 dimensional representation for the tdm :**

```
> tdm_dim_50 <- svd_dec$v[,1:50] %*% Dim_50 %*% t(svd_dec$u[,1:50])
> dim(tdm_dim_50)
[1] 7026 2661
> tdm_dim_100 <- svd_dec$v[,1:100] %*% Dim_100 %*% t(svd_dec$u[,1:100])
> dim(tdm_dim_100)
[1] 7026 2661
> tdm_dim_200 <- svd_dec$v[,1:200] %*% Dim_200 %*% t(svd_dec$u[,1:200])
> dim(tdm_dim_200)
[1] 7026 2661
```

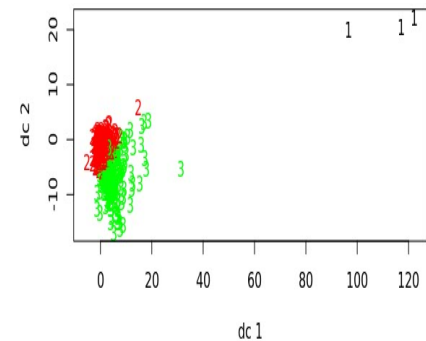
**Clustering the d=50,100,200 dimensional documents and the words with kMeans and as K=3 :**



**D=50**



**D=100**



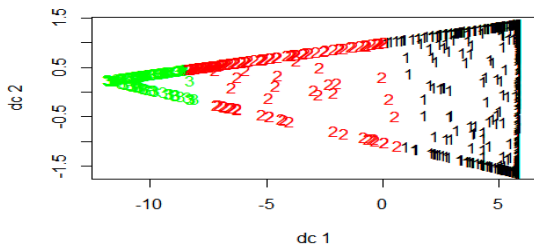
**D=200**

- The SSE value for D=50,100,200 is 879310.22,971192.59,1066057.96 respectively as here we observe that D=50 has the least SSE value hence we can choose this dimension for clustering

```
> print(paste("SSE Value for SVD Dim 50:",clus$totss))
[1] "SSE Value for SVD Dim 50: 879310.222813378"
> print(paste("SSE Value for SVD Dim 100:",clus$totss))
[1] "SSE Value for SVD Dim 100: 971192.587482241"
> print(paste("SSE Value for SVD Dim 200:",clus$totss))
[1] "SSE Value for SVD Dim 200: 1066057.95585525"
```

### 3. LDA - NEWSGROUPS:

- LDA is used to produce the topics or concepts which are related to the documents and terms
- The main parameter to see here is the no. of topics produced by LDA and I got it as 3

| Clustering LDA vectors with kmeans with k=3:   | Top concepts with most representative words:  |             |         |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
|--|---|-------------|---------|---------|---------|------|------------|-------------|---------|------|----------|----------|--------|------|--------|-----------|--------|------|-----------|--------|--------|------|--------|---------|--------|
|   | <pre>&gt; lda_dtm_pruned&lt;-LDA(dtm_pruned,3) &gt; lda_rep_words&lt;-as.matrix(terms(lda_dtm_pruned,5)) &gt; lda_rep_words</pre> <table><thead><tr><th></th><th>Topic 1</th><th>Topic 2</th><th>Topic 3</th></tr></thead><tbody><tr><td>[1,]</td><td>"armenian"</td><td>"christian"</td><td>"drive"</td></tr><tr><td>[2,]</td><td>"israel"</td><td>"weapon"</td><td>"file"</td></tr><tr><td>[3,]</td><td>"isra"</td><td>"firearm"</td><td>"imag"</td></tr><tr><td>[4,]</td><td>"turkish"</td><td>"fire"</td><td>"scsi"</td></tr><tr><td>[5,]</td><td>"arab"</td><td>"jesus"</td><td>"card"</td></tr></tbody></table> <p>\\</p> |             | Topic 1 | Topic 2 | Topic 3 | [1,] | "armenian" | "christian" | "drive" | [2,] | "israel" | "weapon" | "file" | [3,] | "isra" | "firearm" | "imag" | [4,] | "turkish" | "fire" | "scsi" | [5,] | "arab" | "jesus" | "card" |
|  | Topic 1   | Topic 2     | Topic 3 |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| [1,]   | "armenian"  | "christian" | "drive" |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| [2,]   | "israel"  | "weapon"    | "file"  |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| [3,]   | "isra"  | "firearm"   | "imag"  |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| [4,]   | "turkish"   | "fire"      | "scsi"  |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| [5,]   | "arab"  | "jesus"     | "card"  |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |
| <ul style="list-style-type: none"><li>The above figure shows the clustering of the topics with number of clusters as 3</li></ul> | <ul style="list-style-type: none"><li>Here there are 3 concepts with most 5 representative words. E.g. Topic 1 has most 5 representative words as:Armenian,Israel,isra,Turkish,arab</li></ul>   |             |         |         |         |      |            |             |         |      |          |          |        |      |        |           |        |      |           |        |        |      |        |         |        |

- Below is how I got the clustering of LDA vectors by calculating the topic probability by using as.data.frame and then used it to plot the cluster and also got SSE as 1483.04 :

```
> lda_dtm_pruned<-LDA(dtm_pruned, 3)
> topicProb<-as.data.frame(lda_dtm_pruned@gamma)
> head(topicProb)
```

|   | v1           | v2           | v3        |
|---|--------------|--------------|-----------|
| 1 | 0.0003154883 | 0.0003154883 | 0.9993690 |
| 2 | 0.0014105369 | 0.0014105369 | 0.9971789 |
| 3 | 0.0008989978 | 0.0008989978 | 0.9982020 |
| 4 | 0.0009886309 | 0.0009886309 | 0.9980227 |
| 5 | 0.0016444666 | 0.0016444666 | 0.9967111 |
| 6 | 0.0008675361 | 0.1624582465 | 0.8366742 |

```
> clus<-kmeans(topicProb,3)
> plotcluster(topicProb,clus$cluster)
> print(paste("SSE value for LDA(NewsGroups):",clus$totss))
[1] "SSE value for LDA(NewsGroups): 1483.03872127156"
```

## II. B. CLUSTERING EXPERIMENTS-Yelp

- CLUSTERING:** Used Nbclust for indexing with index as duda hence k=3 and then clustered using kmeans and got SSE as 166382.16

```
> yelp_no_clusters$Best.nc
Number_clusters Value_Index
3.0000 31.1909
> yelp_clus<-kmeans(yelp_matrix_pruned_tfidf,3)
> fviz_cluster(yelp_clus, data = matrix_pruned_tfidf, geom = "point", stand = FALSE, frame.type = "norm")
[1] "SSE Value for yelp_matrix_pruned_tfidf : 16382.1558407733"
```

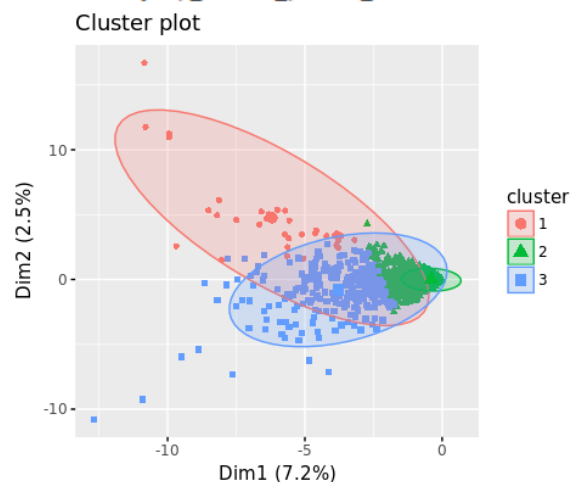


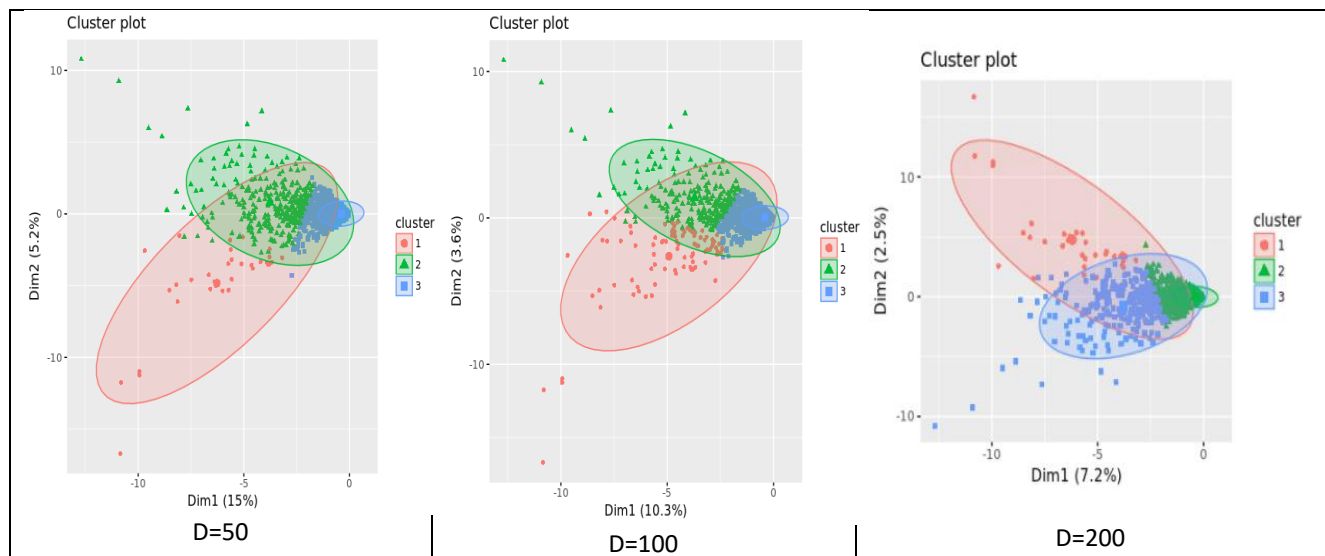


Fig. shows cluster for k=3 using fviz:

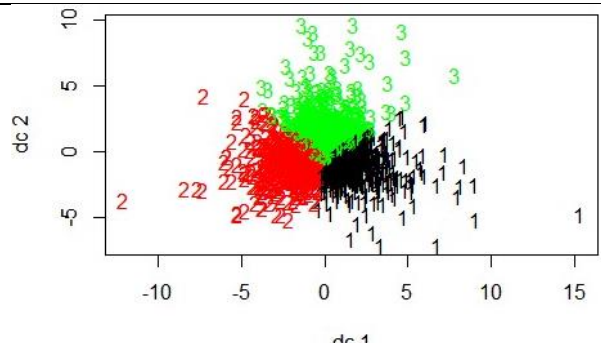
## 2. LSA-YELP:

- The SVD of the dtm was computed as : `yelp_svd_dec <- svd(yelp_dtm_pruned)`
- Computed D=50,100,200 dimensional representation for the tdm by multiplying  $v, d(\text{ford}=50,100,200), u^T$  and dimension of all tdm was 3845\*5000
- Clustered for all d=50,100,200 dimensional documents and the words with kMeans and as K=3 using fviz and got SSE as 50128.32,73659.22,106085.36 respectively ans as D=50 has least SSE so we can choose it

```
> yelp_clus<-kmeans(yelp_tdm_dim_50,3)
> fviz_cluster(yelp_clus, data = yelp_tdm_dim_50, geom = "point", stand = FALSE, frame.type = "norm")
> print(paste("SSE Value for SVD Dim 50(Yelp):",yelp_clus$totss))
[1] "SSE Value for SVD Dim 50(Yelp): 50128.3212584285"
```



## 3. LDA-YELP:

| Clustering LDA vectors with kmeans with k=3:   | Top concepts with most representative words:   |          |          |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
|--|--|----------|----------|---------|---------|------|---------|----------|----------|------|--------|--------|----------|------|--------|--------|--------|------|--------|----------|--------|------|----------|----------|---------|
|  <p>The above figure shows the clustering of the topics with number of clusters as 3 with SSE=3285.36</p> | <pre>&gt; yelp_lda_rep_words</pre> <table><tr><th></th><th>Topic 1</th><th>Topic 2</th><th>Topic 3</th></tr><tr><td>[1,]</td><td>"pizza"</td><td>"actual"</td><td>"person"</td></tr><tr><td>[2,]</td><td>"told"</td><td>"home"</td><td>"someth"</td></tr><tr><td>[3,]</td><td>"sauc"</td><td>"care"</td><td>"taco"</td></tr><tr><td>[4,]</td><td>"shop"</td><td>"flavor"</td><td>"line"</td></tr><tr><td>[5,]</td><td>"dinner"</td><td>"return"</td><td>"excel"</td></tr></table> <p>The most 5 representative words for 3 topics are as follows:<br/>For topic 1: pizza,told,sauc,shop,dinner which are all related to food concept</p> |          | Topic 1  | Topic 2 | Topic 3 | [1,] | "pizza" | "actual" | "person" | [2,] | "told" | "home" | "someth" | [3,] | "sauc" | "care" | "taco" | [4,] | "shop" | "flavor" | "line" | [5,] | "dinner" | "return" | "excel" |
|  | Topic 1  | Topic 2  | Topic 3  |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
| [1,]   | "pizza"  | "actual" | "person" |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
| [2,]   | "told"   | "home"   | "someth" |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
| [3,]   | "sauc"   | "care"   | "taco"   |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
| [4,]   | "shop"   | "flavor" | "line"   |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |
| [5,]   | "dinner"   | "return" | "excel"  |         |         |      |         |          |          |      |        |        |          |      |        |        |        |      |        |          |        |      |          |          |         |

**Analysis of clustering both datasets:** Got k=3 for both and clustering LSA vectors was better for D=50 as it has least SSE. Clustering LDA vectors, observed in yelp cluster the center of clusters are more closer as the words are more generic and can also be used to find the relativeness of the topics.

## (II).(C).(4) Evaluation and (II).(D)Result Summary in table form



| Algorithm Used | # Clusters(K) | SSE – NewsGroups   | SSE - Yelp   |
|----------------|---------------|--|--|
| Kmeans         | 3             | 3243.94  | 16382.16   |
| LSA            | 3             | 879310.22 (D=50 )<br>971192.59 (D=100)<br>1066057.96 (D=200) | 50128.32 (D=50)<br>73659.21 (D=100)<br>106085.36 (D=200) |
| LDA            | 3             | 1483.04  | 3285.36  |

**ACURACY:** The accuracy is found by using the confusion matrix using below code:

```
(sum(apply(con_lda,1,max))/sum(clus$size))*100
```

- Higher the value of it then more accurate the algorithm is.
- Also observed,if the processing of data set and clustering of the data set is complex then the accuracy is low for that algorithm
- The accuracy of yelp data set for each algorithm is less then the one for 20NG because the yelp data was difficult to process and was a challenge.
- For both the data sets the accuracy for LSA algorithm was low for higher dimension (D=200) and was high for low dimension (D=50)
- LDA is better at producing the topics and the best representative terms/words

#### Analysis for both the datasets:

- The above table shows the comparison between the SSE produced for different algorithms kmeans,LSA(D=50,100,200),LDA for both the datasets NewsGroups and Yelp. From my observation from thr table above for both the datasets, the value of SSE is low for LDA for k=3.Hence,LDA will produces better outputs and LSA and Kmeans. LDA clustering is more efficient among these algorithms used as it clusters the document using the concepts or topics

### III. ANALYSIS

The following are the steps that walk you through the assignment , it also discusses about the algorithms like LSA,LDA,kmeans and their performance, and also includes the things I learnt :

- The data analysis and clustering was done on two datasets: 20NewsGroups,Yelp
- Firstly,the corpus of the datasets were made using tm(**text mining**) package and then it was cleaned by removing the English **stop words**,punctuations,numbers,**stemming** of the corpus using tm\_map()
- After that the corpus was **pruned** and **dtm** was made.The pruned dtm resulted in lot of reduction of no. of terms which reduced the processing a lot.
- Then produced the **wordcloud** of the pruned dtm there after using **NbClust** the No. of clusters were found
- **Kmeans** was used for clustering and then after for **LSA,SVD** of tf-idf matrix was formed

and for D=50,100,200 of SVD clustering was performed for them and also observed that increasing the D would not affect the dimensions of the matrix and D=50 has least SSE for both data sets

- Moving ahead, using **LDA** I got the topics in the documents and also found most representative words for each topics

- After observing the SSE for all the three algorithms, it was concluded that LDA would perform better and is more efficient than other two, as LDA had least SSE value for both the datasets

#### IV. LSA DERIVATION

##### a. Goal : To compute $AA^T$ and $A^T A$ using A's SVD and get representation of them with 3 matrices

- Representing  $AA^T$  into 3 matrices** : As we know A's SVD is,  $A=U\Sigma V^T$

and therefore  $A^T = (U\Sigma V^T)^T = V\Sigma^T U^T$

Hence,  $AA^T = (U\Sigma V^T)(V\Sigma^T U^T)$

$$\Rightarrow AA^T = U\Sigma\Sigma^T U^T = U\Sigma^2 U^T$$

Hence, we get  $AA^T = U\Sigma^2 U^T$

Here, U is Eigenvectors of  $AA^T$

And  $\Sigma^2$  is Eigenvalue of  $AA^T$

$\Sigma^2 \rightarrow$  are values of the  $\sigma_i^2$

$\sigma_i^2 \rightarrow$  they are the positive sq. root of  $\lambda_i$

- Representing  $A^T A$  into 3 matrices** : As we know A's SVD is,  $A=U\Sigma V^T$

and therefore  $A^T = (U\Sigma V^T)^T = V\Sigma^T U^T$

Hence,  $A^T A = (V\Sigma^T U^T)(U\Sigma V^T)$

$$\Rightarrow A^T A = V\Sigma^T \Sigma V^T = V\Sigma^2 V^T$$

Hence, we get  $A^T A = V\Sigma^2 V^T$  =====>

$$= V \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix} V^T.$$

Here, V is Eigenvectors of  $A^T A$

And  $\Sigma^2$  is Eigenvalue of  $A^T A$

$\Sigma^2 \rightarrow$  are values of the  $\sigma_i^2$

$\sigma_i^2 \rightarrow$  they are the positive sq. root of  $\lambda_i$

- Relation between SVD of A and Eigenvalue decomposition of  $AA^T$  and  $A^T A$ :**

The A's SVD has diagonal as  $\Sigma$  but  $AA^T$  and  $A^T A$  has diagonal as  $\Sigma^2$  which are values of the  $\sigma_i^2$  and  $\sigma_i^2$  is the positive sq. root of  $\lambda_i$ . Also in A's SVD A's Eigenvector is U where as  $AA^T$  and  $A^T A$  Eigenvectors are U and V respectively.

##### b. R command for LSA document representation using SVD matrices

- Here, first the svd is created of pruned matrix and then by selecting most k terms i.e. for D=50,100,200 LSA document is represented

```
svd_dec <- svd(dtm_pruned)
Diagonal <- diag(svd_dec$d)
Dim_50 <- Diagonal[1:50,1:50]
Dim_100 <- Diagonal[1:100,1:100]
Dim_200 <- Diagonal[1:200,1:200]

> tdm_dim_50 <- svd_dec$v[,1:50] %*% Dim_50 %*% t(svd_dec$u[,1:50])
> dim(tdm_dim_50)
[1] 7026 2661
> tdm_dim_100 <- svd_dec$v[,1:100] %*% Dim_100 %*% t(svd_dec$u[,1:100])
> dim(tdm_dim_100)
[1] 7026 2661
> tdm_dim_200 <- svd_dec$v[,1:200] %*% Dim_200 %*% t(svd_dec$u[,1:200])
> dim(tdm_dim_200)
[1] 7026 2661
```