NATURAL LANGUAGE PROCESSING

TEXT CLUSTERING

TARA JOSEPH

GITAA PVT LTD

TABLE OF CONTENTS

I. PROBLEM STATEMENT

To cluster textual data with Natural Language Processing using clustering techniques after finding the optimal number of clusters.

II. OBJECTIVE

Data as text has been increasing exponentially as a result of social interaction, communication, and culture being largely digitalized. It is usually unstructured as it does not follow any norm. Text Analysis enables corporations, governments, researchers, and the media to make critical decisions based on the vast amounts of data available to them. To enable computers to work with this kind of data, Natural Language Processing (NLP) was introduced.

Natural language processing (NLP) is used in text analysis to convert unstructured text in files into normalized, structured data that can be analyzed and then used to run machine learning (ML) algorithms. *(refer fig 1)*

In this project, fundamental text analytical tasks, including text preprocessing and text document clustering algorithm, are applied. Clustering methods are unsupervised algorithms that aid in the summarization of information from large amounts of text data, by forming various groups. This method is beneficial for determining what the dataset is primarily about and how you might categorize the context of the text into distinct groups.

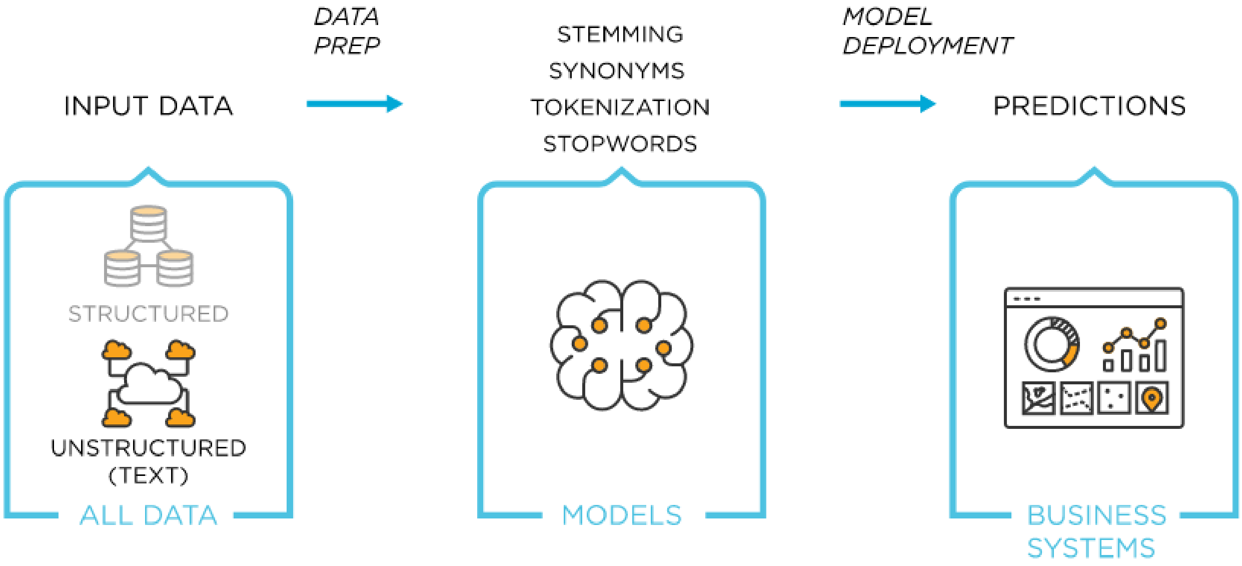
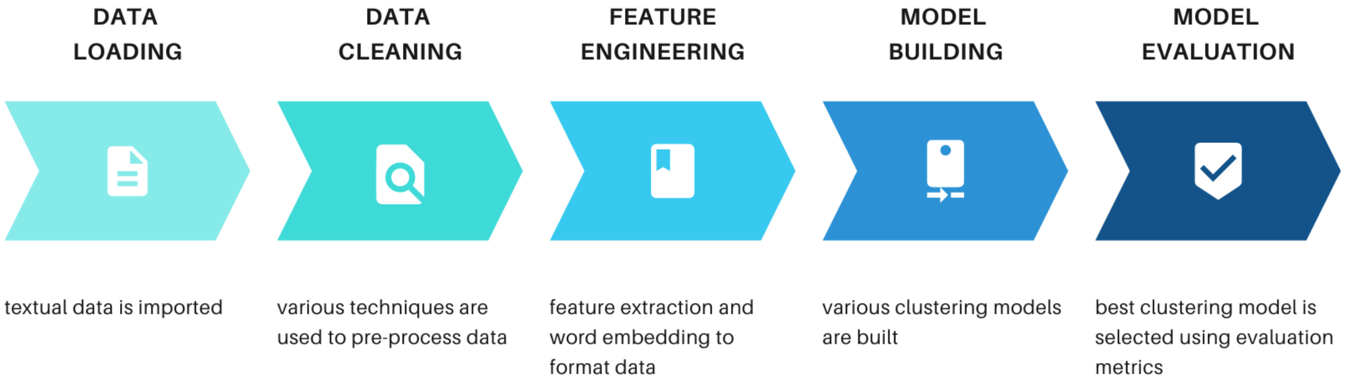


fig 1. Text Analysis procedure

The main motivation of this project was to effectively cluster the text data with the optimal number of clusters and build the best working model on them. Different techniques to pre-process the data were used and various clustering algorithms were introduced and evaluated.NLP techniques such as stemming and lemmatization were examined with different text representations. In NLP, feature extraction and word embedding techniques are trivial in formatting the data. Also known as text vectorization, the data is converted into a numerical form for the machines to understand it.

A crucial step in text clustering is identifying the optimal number of clusters that the text can be categorized into. Estimating the optimal number of clusters was done using the Depth Difference method (DeD) proposed by Channamma Patil and Ishwar Baidari. The DeD method estimates the *cluster number* before actual clustering is constructed. To achieve accurate clustering results, the number of clusters defined must be accurate since the results may largely depend on it. The proposed Ded method was proved to outperform the other cluster number estimation method.

III. Text Clustering Pipeline



DATA LOADING
textual data is imported

DATA CLEANING
various techniques are used to pre-process data

FEATURE ENGINEERING
feature extraction and word embedding to format data

MODEL BUILDING
various clustering models are built

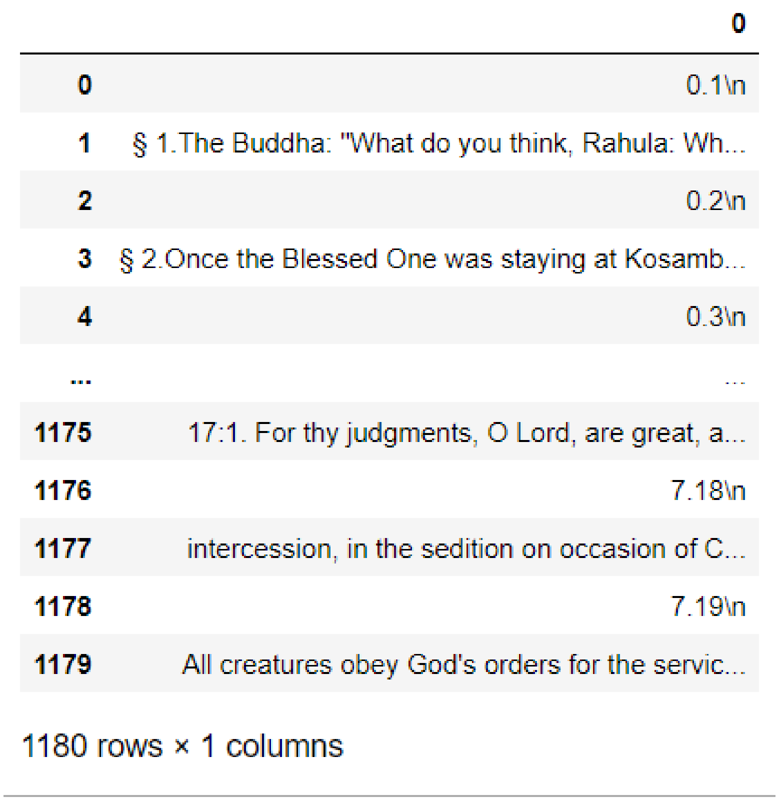
MODEL EVALUATION
best clustering model is selected using evaluation metrics

1. DATA LOADING

The textual data provided here consists of various religious texts which need to be clustered based on the book it belongs to.

The data was imported onto the Jupyter notebook after which it was converted into a data frame for it to be easily analyzed.

Input data frame:



	0
0	0.1\n
1	§ 1.The Buddha: "What do you think, Rahula: Wh...
2	0.2\n
3	§ 2.Once the Blessed One was staying at Kosamb...
4	0.3\n
...	...
1175	17:1. For thy judgments, O Lord, are great, a...
1176	7.18\n
1177	intercession, in the sedition on occasion of C...
1178	7.19\n
1179	All creatures obey God's orders for the servic...

1180 rows × 1 columns

fig 2. Input data frame consisting of 1180 rows

2. DATA CLEANING

Cleaning textual data known as text preprocessing involves transforming text into a clean and consistent format that can then be fed into a model for further analysis and evaluation. This step is crucial since it improves the efficiency of these algorithms. There are various techniques that clean text according to the application it caters to. However, tokenization, filtering, and normalization, which includes stemming or lemmatization is usually involved in processing any textual data.

In this data set, the following techniques were applied to preprocess the text.

i) Lowercasing the text
All the words in the text are converted to lowercase to maintain uniformity. This also prevents the machine from considering the same words written in different cases as different entities.

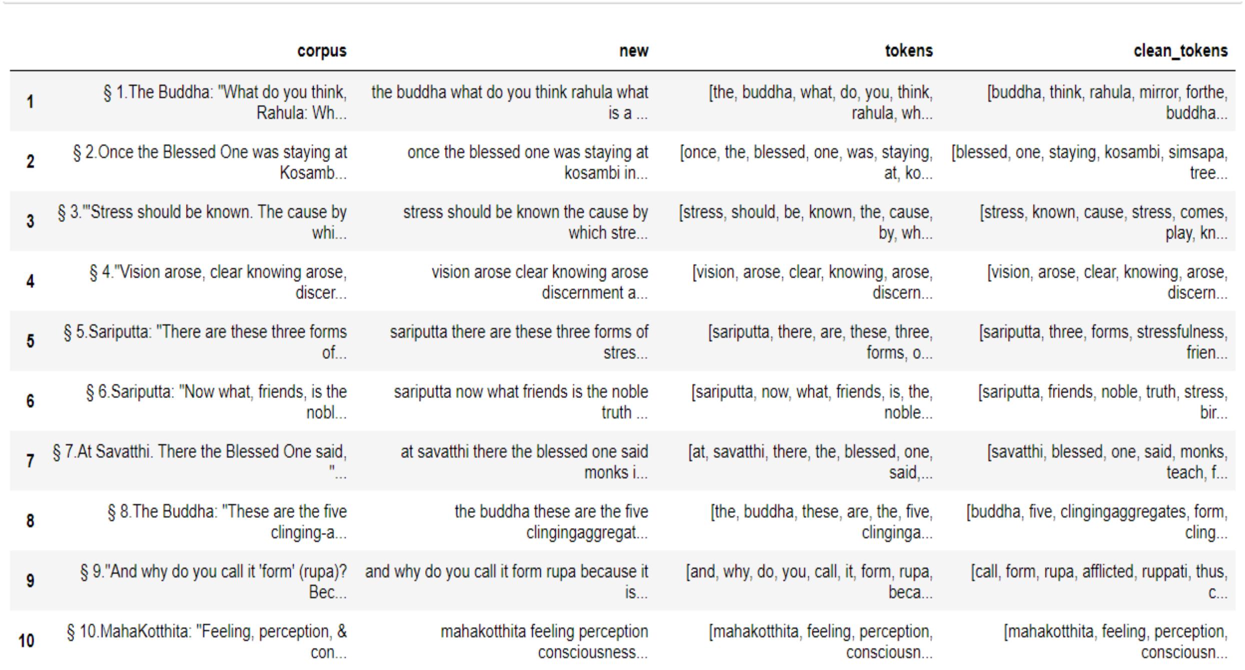
ii) Data filtering
This involves the removal of punctuations, digits, whitespace, symbols, and HTML tags that are not required for clustering purposes. The rows with missing input were also removed.

After this step, there were 589 rows with 2 columns in the data frame where the filtered and lowercased text was stored under column ‘new’.

iii) Tokenization
The next preprocessing step was to tokenize the text. In this case, each sentence was segregated into smaller units such as words. Each unit is then considered as a token. Tokenization's basic premise is to try and decipher the meaning of a text by analyzing the smaller units or tokens that constitute a sentence.
To do this, a function was defined to tokenize the text after which it was applied to the filtered data. The Natural Language Toolkit(NLTK) library in python is also largely used for Text Preprocessing.

iv) Stop word removal
Another method that involves filtering out text is stop word removal. It is used to eliminate often used terms that don't contain much information and are mostly used for grammatical purposes. These words are called stopwords such as *a, an, the, her*self, etc.

The cleaned tokens that are derived after this step is stored under the column name ‘clean_tokens’. *(refer fig.3)*



	corpus	new	tokens	clean_tokens
1	§ 1.The Buddha: "What do you think, Rahula: Wh...	the buddha what do you think rahula what is a ...	[the, buddha, what, do, you, think, rahula, wh...	[buddha, think, rahula, mirror, forthe, buddha...
2	§ 2.Once the Blessed One was staying at Kosamb...	once the blessed one was staying at kosambi in...	[once, the, blessed, one, was, staying, at, ko...	[blessed, one, staying, kosambi, simsapa, tree...
3	§ 3."'Stress should be known. The cause by whi...	stress should be known the cause by which stre...	[stress, should, be, known, the, cause, by, wh...	[stress, known, cause, stress, comes, play, kn...
4	§ 4."Vision arose, clear knowing arose, discer...	vision arose clear knowing arose discernment a...	[vision, arose, clear, knowing, arose, discern...	[vision, arose, clear, knowing, arose, discern...
5	§ 5.Sariputta: "There are these three forms of...	sariputta there are these three forms of stres...	[sariputta, there, are, these, three, forms, o...	[sariputta, three, forms, stressfulness, frien...
6	§ 6.Sariputta: "Now what, friends, is the nobl...	sariputta now what friends is the noble truth ...	[sariputta, now, what, friends, is, the, noble...	[sariputta, friends, noble, truth, stress, bir...
7	§ 7.At Savatthi. There the Blessed One said, "...	at savatthi there the blessed one said monks i...	[at, savatthi, there, the, blessed, one, said,...	[savatthi, blessed, one, said, monks, teach, f...
8	§ 8.The Buddha: "These are the five clinging-a...	the buddha these are the five clingingaggregat...	[the, buddha, these, are, the, five, clinginga...	[buddha, five, clingingaggregates, form, cling...
9	§ 9."And why do you call it 'form' (rupa)? Bec...	and why do you call it form rupa because it is...	[and, why, do, you, call, it, form, rupa, beca...	[call, form, rupa, afflicted, ruppati, thus, c...
10	§ 10.MahaKotthita: "Feeling, perception, & con...	mahakotthita feeling perception consciousness...	[mahakotthita, feeling, perception, consciousn...	[mahakotthita, feeling, perception, consciousn...

fig. 3 Various stages of text preprocessing until normalization

v) Stemming

Stemming is the process of reduction of a word into its root or stem word. The word affixes are removed leaving behind only the root form.

We do this by importing PorterStemmer from nltk and make a class for it. After that using the PorterStemmer object we call the stem method to perform stemming on our word list.

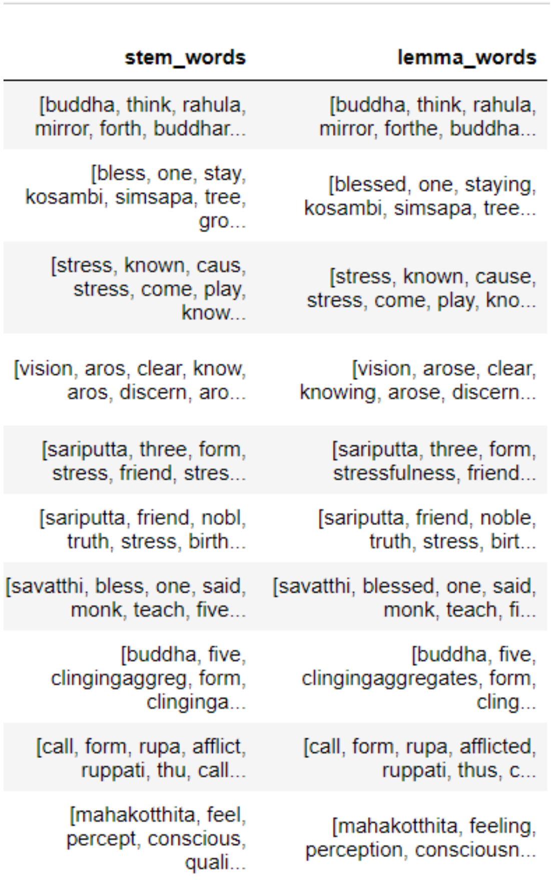
vi) Lemmatization

Lemmatization is a method for combining various inflected forms of words into a single root form with the same meaning. It's comparable to stemming because it results in a stripped-down word with dictionary meaning.

Similar to stemming, WordNetLemmatizer imported from nltk is used to reduce sentences to lemma words.

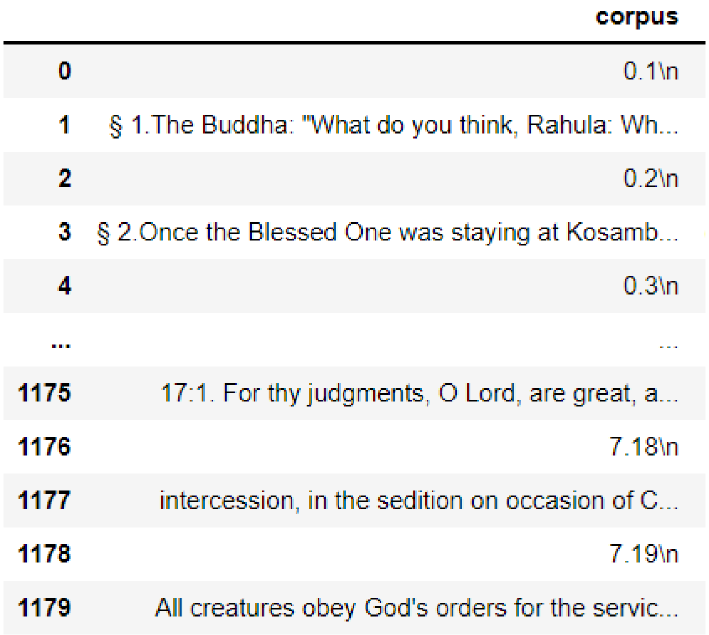
In this case, we can see that stemming gave better results for bringing a word to its root form.

Hence, we progress with the use of stem words to generate a new text by combining the words which would act as input for further analysis. The final regenerated text from the root words has been reduced to 589 rows since, in the original data frame, every alternate line had only unwanted symbols and numbers which were dropped. (refer fig. 4).

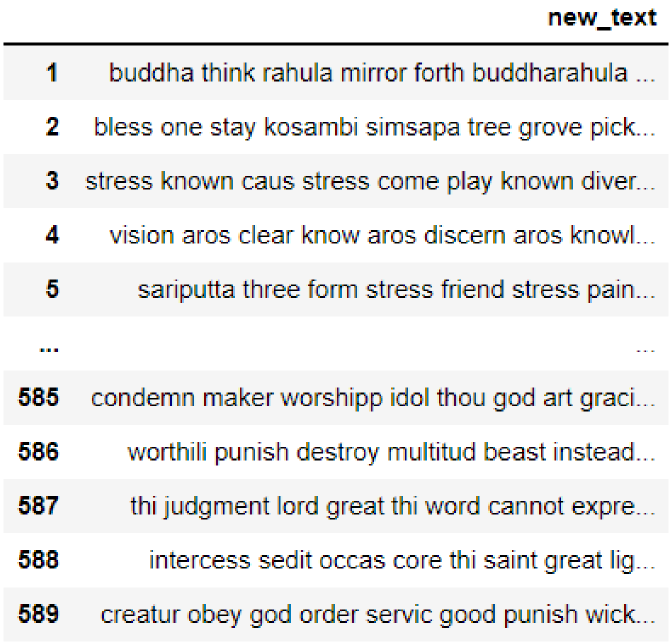


stem_words	lemma_words
[buddha, think, rahula, mirror, forth, buddhar...	[buddha, think, rahula, mirror, forthe, buddha...
[bless, one, stay, kosambi, simsapa, tree, gro...	[blessed, one, staying, kosambi, simsapa, tree...
[stress, known, caus, stress, come, play, know...	[stress, known, cause, stress, come, play, kno...
[vision, aros, clear, know, aros, discern, aro...	[vision, arose, clear, knowing, arose, discern...
[sariputta, three, form, stress, friend, stres...	[sariputta, three, form, stressfulness, friend...
[sariputta, friend, nobl, truth, stress, birth...	[sariputta, friend, noble, truth, stress, birt...
[savatthi, bless, one, said, monk, teach, five...	[savatthi, blessed, one, said, monk, teach, fi...
[buddha, five, clingingaggreg, form, clinginga...	[buddha, five, clingingaggregates, form, cling...
[call, form, rupa, afflict, ruppati, thu, call...	[call, form, rupa, afflicted, ruppati, thus, c...
[mahakotthita, feel, percept, conscious, quali...	[mahakotthita, feeling, perception, consciousn...

fig.4 Preprocessed text with original data frame



	corpus
0	0.1\n
1	§ 1.The Buddha: "What do you think, Rahula: Wh...
2	0.2\n
3	§ 2.Once the Blessed One was staying at Kosamb...
4	0.3\n
...	...
1175	17:1. For thy judgments, O Lord, are great, a...
1176	7.18\n
1177	intercession, in the sedition on occasion of C...
1178	7.19\n
1179	All creatures obey God's orders for the servic...



	new_text
1	buddha think rahula mirror forth buddharahula ...
2	bless one stay kosambi simsapa tree grove pick...
3	stress known caus stress come play known diver...
4	vision aros clear know aros discern aros knowl...
5	sariputta three form stress friend stress pain...
...	...
585	condemn maker worshipp idol thou god art graci...
586	worthili punish destroy multitud beast instead...
587	thi judgment lord great thi word cannot expre...
588	intercess sedit occas core thi saint great lig...
589	creatur obey god order servic good punish wick...

3. FEATURE ENGINEERING

> ***"If 80 percent of our work is data preparation, then ensuring data quality is the important work of a machine learning team." – Andrew Ng***

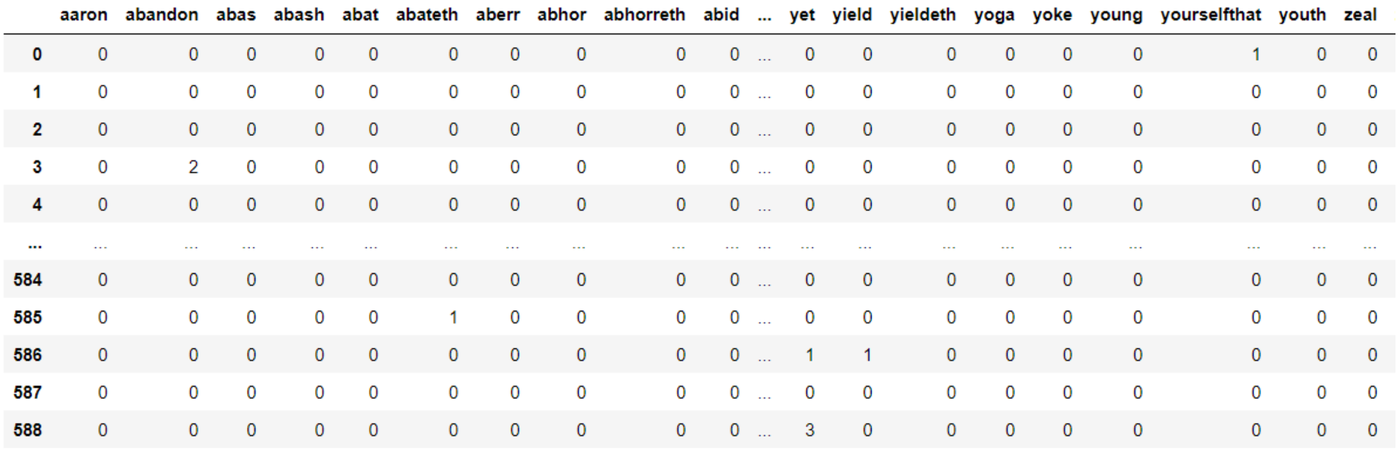
Feature engineering is one of the most significant steps in machine learning. The importance of feature engineering is even more important for unstructured, textual data because we need to represent it numerically for it to be understood by machine learning algorithms to produce reasonable outcomes. Frequently used strategies to achieve this includes Bag Of Words method, Count Vectorization, TF-IDF, One Hot Encoding, etc.

❖ Count Vectorization

This involves mapping text data into a vector form. This process is known as vectorization. One of the simplest ways to vectorize is through the word-count vectorizer. It uses a "bag-of-words" approach to handle the text data. It works by transforming the text into vectors on the basis of the frequency of each word in the text.

This was performed using Scikit-learn's built-in CountVectorizer function.

The result we get is shown in fig. 5 below.



	aaron	abandon	abas	abash	abat	abateth	aberr	abhor	abhorreth	abid	...	yet	yield	yieldeth	yoga	yoke	young	yourselfthat	youth	zeal
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	2	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
584	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
585	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0
586	0	0	0	0	0	0	0	0	0	0	...	1	1	0	0	0	0	0	0	0
587	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
588	0	0	0	0	0	0	0	0	0	0	...	3	0	0	0	0	0	0	0	0

fig. 5 CountVectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. In a data frame of shape (589, 5694).

❖ TF-IDF model

TF-IDF stands for Term Frequency Inverse Document Frequency of records. Unlike count vectorization which assigns a higher weight to frequently occurring words, TF-IDF asserts more weight to less frequently occurring words on the assumption that they are more important. It uses two metrics to compute:

- ➔ Term Frequency (TF) - measures how often a term occurs in a given dataset.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

- ➔ Inverse document frequency (IDF) - measures the importance of the term across a corpus.

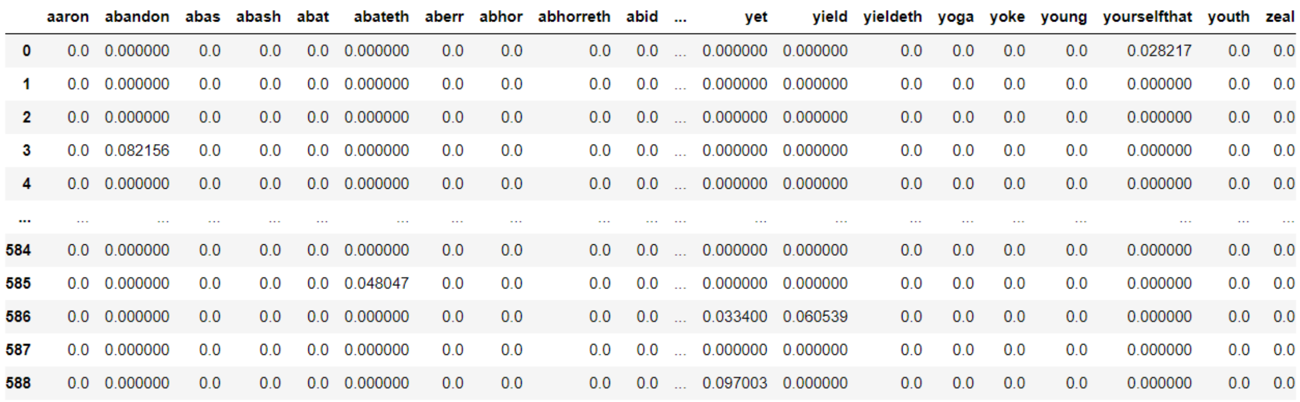
$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

$$TF\text{-}IDF = TF * IDF$$

This type of text representation was done using Scikit-learn’s inbuilt TfidfVectorizer function. The matrix was stored in a data frame with shape (589, 5694).

The output for the same is shown in fig,6 below.



	aaron	abandon	abas	abash	abat	abateth	aberr	abhor	abhorreth	abid	...	yet	yield	yieldeth	yoga	yoke	young	yourselfthat	youth	zeal
0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.028217	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
2	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
3	0.0	0.082156	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
4	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
584	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
585	0.0	0.000000	0.0	0.0	0.0	0.048047	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
586	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.033400	0.060539	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
587	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
588	0.0	0.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.097003	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0

fig. 6 TF-IDF vectorized data

PRINCIPAL COMPONENT ANALYSIS

PCA is a dimension reduction technique that is frequently used for the analysis of high dimensional data from NLP. Word embedding of text data results in matrices that are of very large sizes as seen earlier. However, this makes the data difficult to train and it will be time consuming. Therefore it is important to reduce the number of features to make further NLP analysis more efficient. Applying PCA gives a linear combination of variables which gives maximum variance from the variables. For our data, a variance of 95% was defined.

PCA on count vectorized data gave shape (589,268)

PCA on TD-IDF vectorized data gave shape (589,483).

4. MODEL BUILDING

DETERMINING THE OPTIMAL NUMBER OF CLUSTERS ‘K’

K Means Elbow plot method

K Means is an unsupervised machine learning algorithm that groups data into *k* number of clusters. The number of clusters is pre-defined by the user and the algorithm will try to cluster the data points even if ‘k’ is optimal or not. Therefore, the elbow method is a technique that helps in finding the optimal number of clusters.

The concept is to use k-means clustering for a set of k clusters (let's say 1 to 10) and calculate the sum of squared distances from each point to its assigned center for each value (distortions). When the distortions are plotted and the plot looks like an arm then the “elbow”(the point of inflection on the curve) is the best value of *k*.

However, when this metric was performed, the elbow plot on the count vectorized data, as well as the tf-idf transformed data, gave an undesirable number of clusters.(refer fig. 7 below)

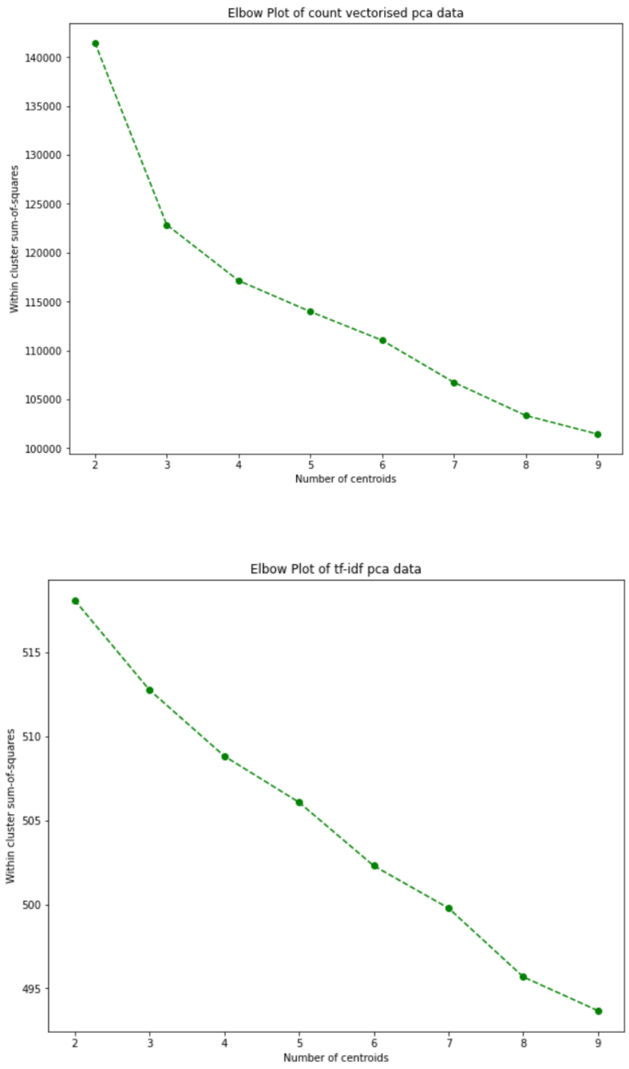


Fig 7

Depth Difference method

As mentioned earlier, the Depth Difference (Ded) method proposed by Channamma Patil and Ishwar Baidari proved to be effective in determining the most optimal number of clusters that text data could be segregated into.

This method determines the k parameter before the data is fitted into a clustering algorithm using data depth. With the help of depth within clusters, depth between clusters, and depth difference we were able to finalize the optimal value of k.

METHOD PROCESS

- **Data Depth**

  The degree of centrality of data is structured using a depth function. The data depth measures the median in the data set which is regarded as the deepest point in the dataset and can be measured using various methods such as convex-hull peeling depth, simplicial depth, regression depth, etc.

  In this case, we use Mahalonobis depth to measure the centrality of the data set. It assigns a value between 0 to 1 to each data point which denotes the deepness of that point in the data set. The point with maximum depth will be the deepest point in the dataset.

  The Mahalanobis depth function is defined as follows:

  $$M_D(x; X) = [1 + (x - \bar{x})^T Cov(X)^{-1}(x - \bar{x})]^{-1}$$

  where,$\bar{x}$ and Cov(X) are the mean and covariance matrix of X, respectively.

  Since the mean is sensitive to outliers, the equation is modified as follows:

  $$M_D(x; X_i) = [1 + (x - X_i)^T Cov(X)^{-1}(x - X_i)]^{-1}$$

  where, each point Xi can be considered a center point, allowing data depth to be calculated from each point in relation to a particular dataset.

➔ DeD Method

Let X = {x1, x2 ...xn} be a dataset with n instances.The depth of each point xi in X is calculated using the above equation and is denoted by Di, for i = 1, 2, ...n.

Depth Median (DM)

Depth median is the deepest point in the dataset X where it has the maximum depth.

$$DM = \max(Di)$$

Depth within cluster (DW)

The depth of each point within a cluster C_k , for k = 2, 3, ...20, is denote by D^k_i
for i = 1, 2, ...n_k , where n_k is the number of points within cluster C_k .
Depth median for each cluster is represented as DM_k

$$DM_k = \max(D_i^k)$$

The average difference between the depths of points within the cluster C_k and the depth median of C_k is denoted by $\triangle_k$, given by the formula

$$\triangle^k = \frac{1}{n_k} \sum_{i \in C_k} |(D_i^k - DM^k)|$$

DW is defined as the average of $\triangle k$ of k clusters as follows:

$$\mathrm{DW} = \frac{1}{k} \sum_{i=1}^{k} (\triangle^i)$$

Depth between clusters (DB)

The average difference between the depths of points within the dataset X and the depth median of X with n as number of instances in data set X is given by:

$$\triangle = \frac{1}{n} \sum_{i=1}^{n} |(D_i - DM)|$$

DB is defined as the difference between △ and DW :

$$DB = \triangle - DW$$

Depth Differrence(DeD)

It is calculated as the difference between DW and DB :

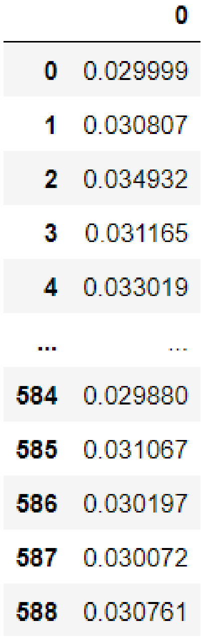
$$DeD = DW - DB$$

Optimal k

The optimal k is the maximum index value of DeD.

$$k = index(max(DeD)$$

Using DistanceMetric function from the sklearn package in Python, the Mahalanobis depth function was computed. The function was called on the tf-idf vectorized data set which had been transformed using PCA. The resulting values were stored in data frame ‘pca_mahal’ where we can see the values ranging between 0 and 1. (refer fig.8)

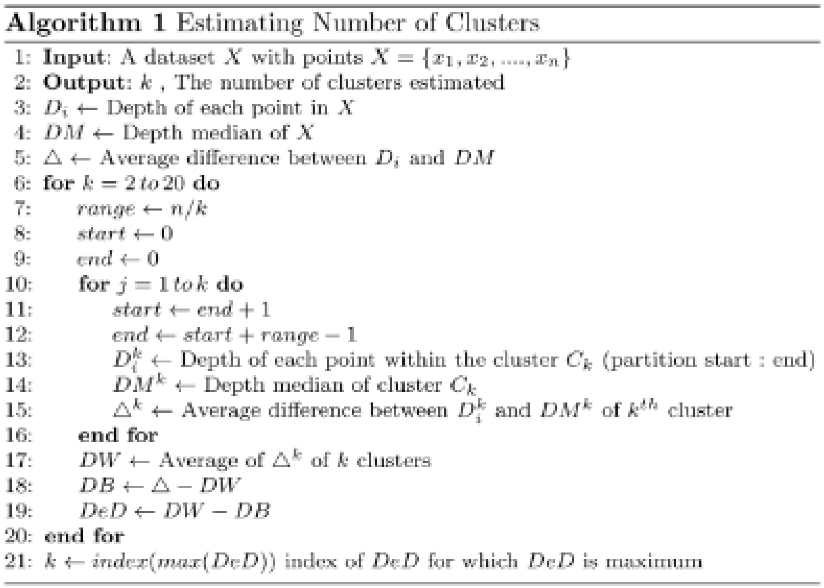


	0
0	0.029999
1	0.030807
2	0.034932
3	0.031165
4	0.033019
...	...
584	0.029880
585	0.031067
586	0.030197
587	0.030072
588	0.030761

Fig 8 pca_mahal data frame consists of depth values.

The Depth Difference method was implemented with respect to the algorithm defined.

After calculating the depth median (DM) of the vector D_i of the dataset X and the average diference between the depths of points within X and DM,the data set was partitioned into k partitions for k=2,....10.Each partition represents one cluster C_k where its depth median is cand depth of each point.With this, DW and DB is computed whose difference gives the depth difference.



```
Algorithm 1 Estimating Number of Clusters
Input: A dataset $X$ with points $X = \{x_1, x_2, ...., x_n\}$
Output: $k$ , The number of clusters estimated
$D_i \leftarrow$ Depth of each point in $X$
$DM \leftarrow$ Depth median of $X$
$\triangle \leftarrow$ Average difference between $D_i$ and $DM$
for $k = 2\ to\ 20$ do
    $range \leftarrow n/k$
    $start \leftarrow 0$
    $end \leftarrow 0$
    for $j = 1\ to\ k$ do
        $start \leftarrow end + 1$
        $end \leftarrow start + range - 1$
        $D_i^k \leftarrow$ Depth of each point within the cluster $C_k$ (partition start : end)
        $DM^k \leftarrow$ Depth median of cluster $C_k$
        $\triangle^k \leftarrow$ Average difference between $D_i^k$ and $DM^k$ of $k^{th}$ cluster
    end for
    $DW \leftarrow$ Average of $\triangle^k$ of $k$ clusters
    $DB \leftarrow \triangle - DW$
    $DeD \leftarrow DW - DB$
end for
$k \leftarrow index(max(DeD))$ index of $DeD$ for which $DeD$ is maximum
```

After obtaining the depth difference values, the index of the maximum value of DeD as the optimal value or 'k' is found by plotting a graph between the number of clusters and each depth difference value.(refer fig 9)

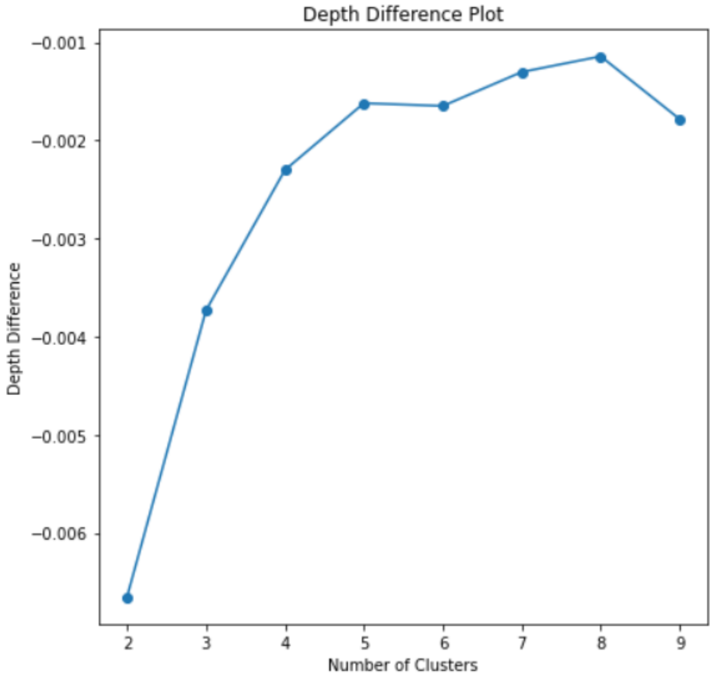


Fig 9 Depth difference plot which gives the optimal number of clusters 'k'

We examine the plot to get 8 as the optimal number of clusters achieved through the DeD method. Using this number as our parameter, we build various clustering models after which their performance is verified using indices such as CH index and DB index.

K Means Clustering

It is a type of unsupervised learning method, which is used for unlabeled data as in our case. In this algorithm, the data were grouped based on high similarity points clustering together and low similarity points in separate clusters.

We make use of the KMeans function in sklearn to fit and transform our data.The plot in fig.10 shows how the data points have been clustered.

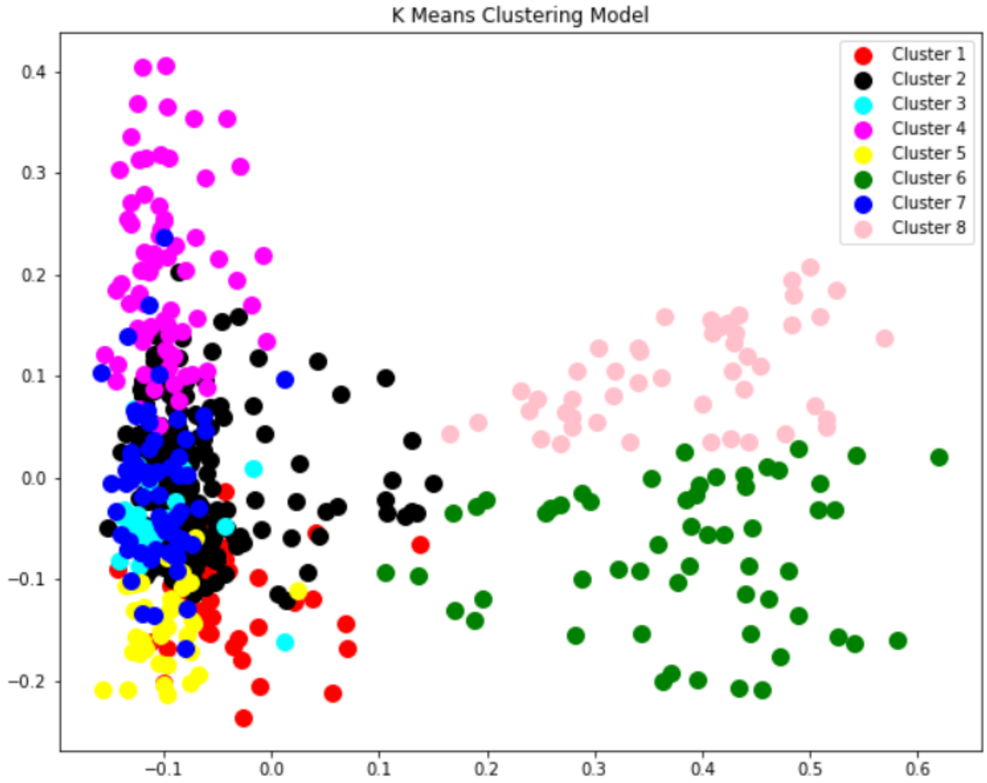


Fig 10 Graphical representation of K means clustered data points on tf-idf transformed data set.

We choose the tf-idf vectorized data over the count vectorized data since the clustering results were more accurate and it worked more efficiently whereas overlapping of the different clusters was largely observed in the count vectorized data.

Hierarchical clustering

It determines cluster assignments by building a hierarchy. Like K-means clustering, hierarchical clustering also groups together the data points with similar characteristics in an unlabeled data set. A bottom-up or top-down technique is used to accomplish this.

- **Agglomerative clustering** is the bottom-up approach. It merges the two most similar points until all of the points have been combined into a single cluster.
- **Divisive clustering** is the top-down approach. It begins with one big cluster with all the data points and splits the least similar clusters to make several smaller clusters.

In our case, we use agglomerative clustering on our tf-idf vectorized data after defining our cluster number as 8.

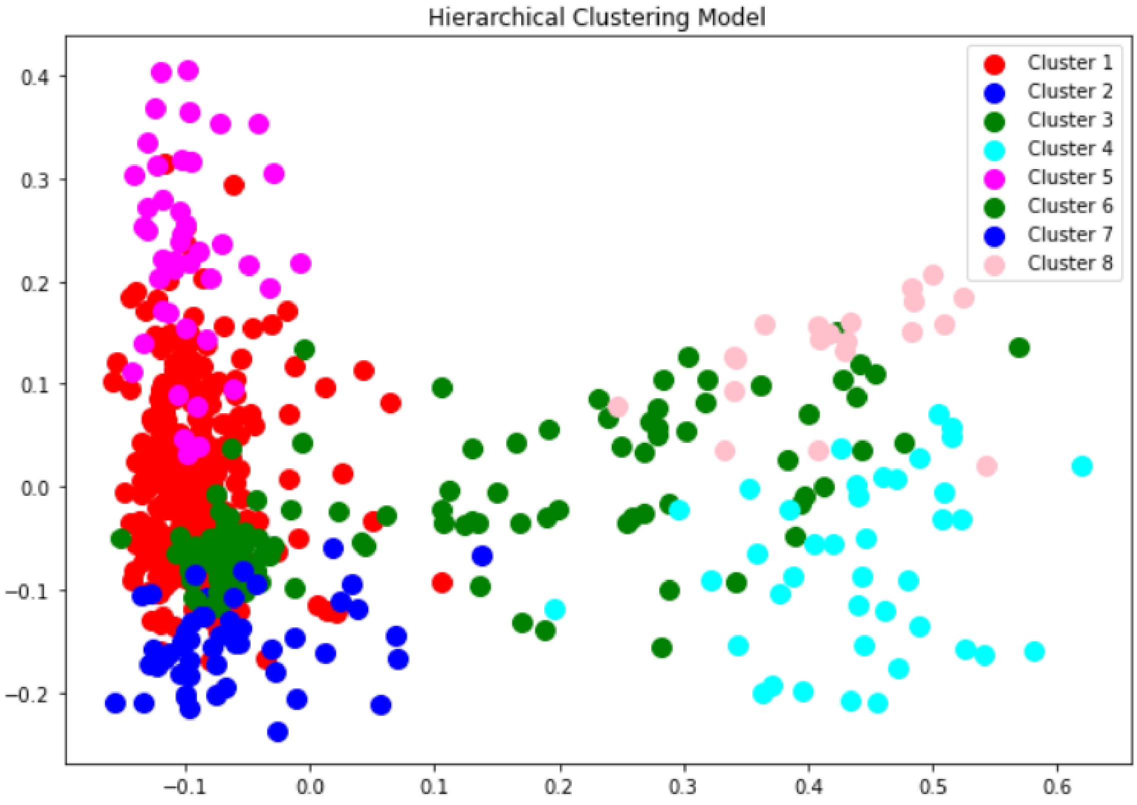


Fig.11. Hierarchical clustering done over tf-idf PCA data set

Spectral Clustering

The usage of spectral clustering has been increasing over the years due to its efficiency in building an accurate model to cluster text data. Standard linear algebra software can solve it quickly, and it frequently outperforms classic techniques like the k-means algorithm.

We defined the model by using Spectral Clustering function from sklearn, then we fitted it to the tf-idf data. This algorithm requires the number of clusters so we set 8 to n_cluster parameter.

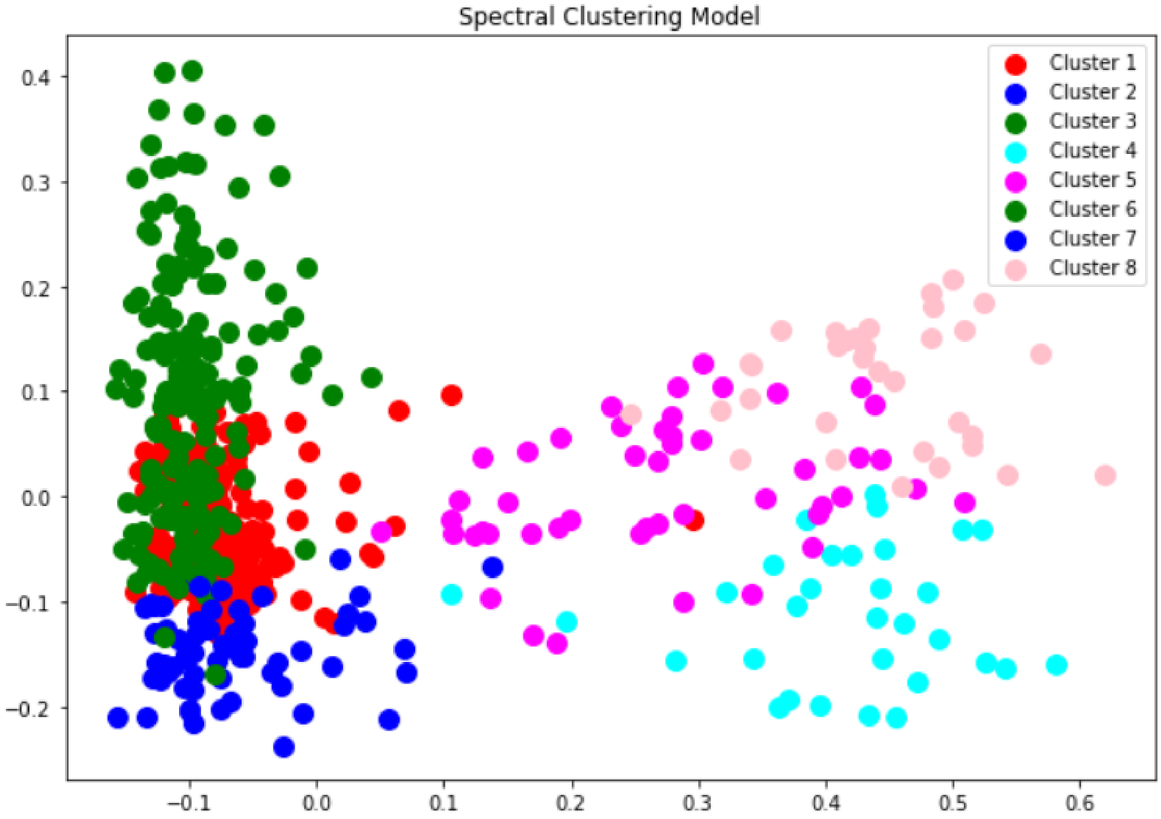


Fig.12. Spectral Clustering on tf-idf vectorized data.

5. MODEL EVALUATION

After clustering our data with different unsupervised algorithms, we evaluate the partitions produced by them by the process called cluster validation. This is an important step in cluster analysis which helps us to determine the algorithm that clustered the data most efficiently. The common approach for this evaluation is to use validity indices.
The internal validity indices that have been proposed to evaluate the models are Calinski-Harabasz (CH) Index and Davies Bouldin (DB) Index.

Clustering Model	CH Index	DB Index
K Means	417.22	6.44
Hierarchical	529.76	5.36
Spectral	760.63	5.50

Higher the CB index, the better the model. However,for DB Index, the lower the average similarity is,the better the clusters are divided and the better the clustering result is.In our case, Spectral clustering seems to give the best results compared to the other clustering algorithms based on the high CH index score as well as a comparatively lower DB index score.

WORD CLOUD

Word clouds are a type of text visualization technique that is commonly used to visualize tags or keywords. Each word in this is represented in a different font size and color. This helps in identifying prominent words since those words will have a larger, bolder font.

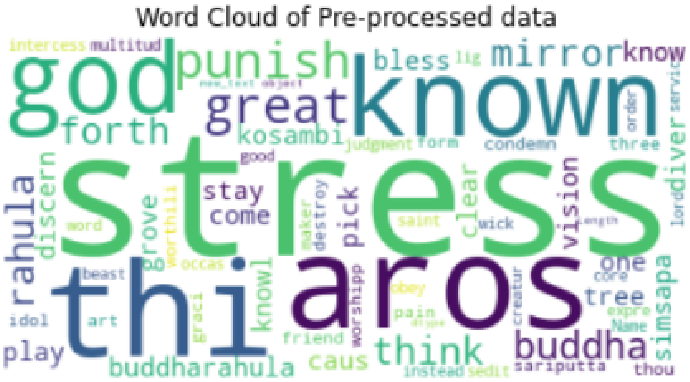
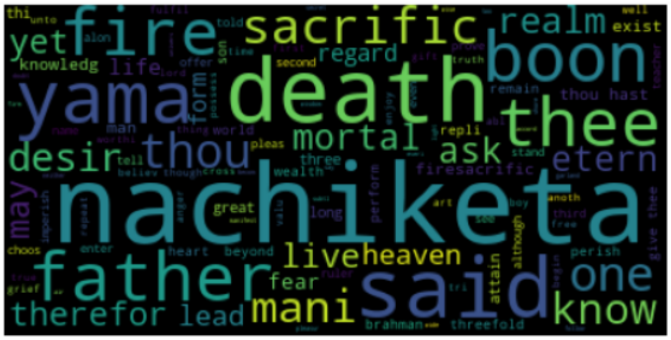
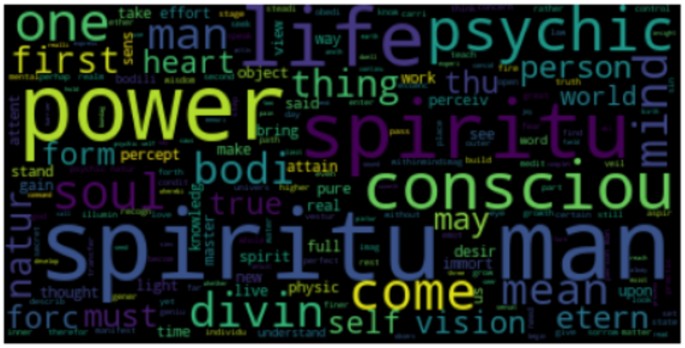


Fig 13 Word Cloud after Preprocessing text

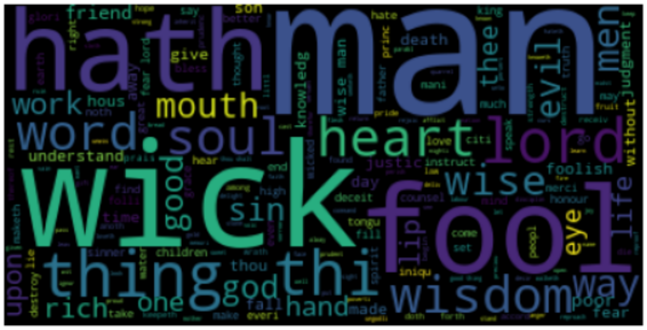
Word clouds are built on each of the eight clusters partitioned by the Spectral Clustering algorithm as it was found to be our most accurate working model.After examining their prominent features ,we were able to label them.The label data is given here.



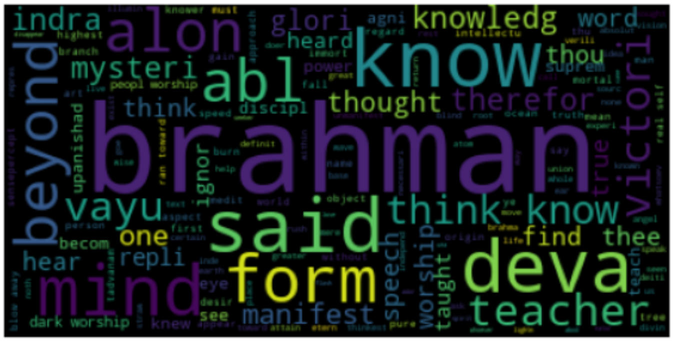
Upanishad



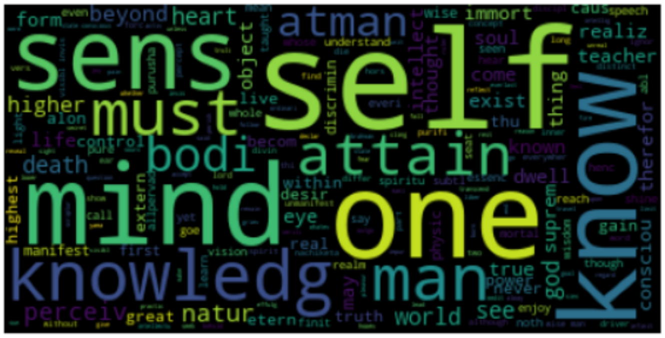
Book of Ecclesiasticus



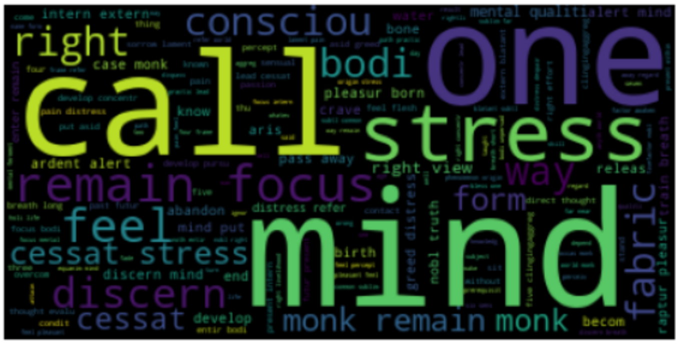
Book of Wisdom



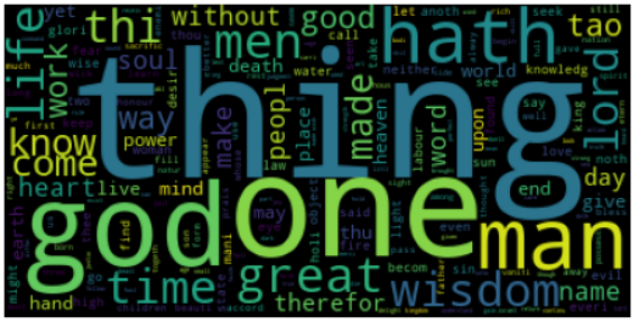
Yoga Sutra



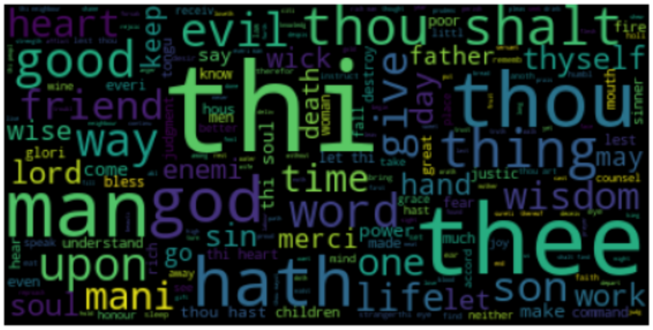
Book of Proverbs



Buddhism



Tao Te Ching



Book of Ecclesiastes

VI. CONCLUSION

The religious text data set was initially preprocessed to improve data quality using several NLP techniques. To map the pre-processed textual data to a numerical representation,tf-idf vectorization gave the most favorable results. Based on these numerical representations the data similarity could be determined, after reducing its dimension using PCA, we were able to obtain the optimal number of clusters through the Ded method which overpowered the K means elbow plot method. Using validity indices we were able to evaluate the clustering algorithms performed. Using Spectral clustering algorithm, we were able to effectively partition the text data to its eight labels of religious books.

V. REFERENCES

Data Preprocessing techniques
https://www.analyticsvidhya.com/blog/2021/09/essential-text-pre-processing-techniques-for-nlp/

Feature Engineering
https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41

Estimation of optimal ‘k’ using DeD method
https://www.researchgate.net/publication/333912095_Estimating_the_Optimal_Number_of_Clusters_k_in_a_Dataset_Using_Data_Depth

Text Document Clustering
https://dc.uwm.edu/cgi/viewcontent.cgi?article=3354&context=etd

David Bouldin Index
https://python-bloggers.com/2021/06/davies-bouldin-index-for-k-means-clustering-evaluation-in-python/

****