PLSA - Text Document Clustering

Name - Amith Korada

Course - AI & ML
(Batch - 4)

Duration - 12 Months

Problem Statement - Perform Hierarchical Clustering from scratch and also using sklearn to perform wholesale customer segmentation based on their annual spending on products

Prerequisites -

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has the latest version of python. The following URL https://www.python.org/downloads/ can be referred to as download python.

The second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this URL https://www.anaconda.com/download/ You will also need to download and install the below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run the below commands in command prompt/terminal to install these packages pip install -U sci-kit-learn pip install NumPy pip install scipy if you have chosen to install anaconda then run the below commands in anaconda prompt to install these packages conda install -c sci-kit-learn conda install -c anaconda numpy conda install -c anaconda scipy.

1. Importing necessary libraries-

```
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.datasets import fetch_20newsgroups
```

2. Loading the dataset-

```
data, _ = fetch_20newsgroups(shuffle=True, random_state=1,
                             remove=('headers', 'footers', 'quotes'),
                             return_X_y=True)
```

3. Convert the collection of text documents to a matrix of token counts-

```
tf_vectorizer = CountVectorizer(max_df=0.95, min_df=2,
                                max_features=n_features,
                                stop_words='english')

tf = tf_vectorizer.fit_transform(data)
```

4. Fitting the LDA model-

```
lda = LatentDirichletAllocation(n_components=num_topics, max_iter=5,
                                learning_method='online',
                                learning_offset=50.,
                                random_state=0)
lda.fit(tf)
```

5. Topics-

```
def get_topics(model, n_top_words, num_topics):
    feat_names = tf_vectorizer.get_feature_names()

    word_dict = {}
    for i in range(num_topics):
        words_ids = model.components_[i].argsort()[:-n_top_words - 1:-1]
        words = [feat_names[key] for key in words_ids]
        word_dict['Topic # ' + '{:02d}'.format(i+1)] = words

    return pd.DataFrame(word_dict)
```

```
import pandas as pd
get_topics(lda, n_top_words, num_topics)
```

	Topic # 01	Topic # 02	Topic # 03	Topic # 04	Topic # 05	Topic # 06	Topic # 07	Topic # 08	Topic # 09	Topic # 10
0	people	government	space	key	edu	god	windows	ax	just	10
1	gun	people	program	car	file	people	use	max	don	00
2	armenian	law	output	chip	com	does	drive	b8f	like	25
3	armenians	mr	entry	used	available	jesus	thanks	g9v	think	15
4	war	use	data	keys	mail	say	does	a86	know	12
5	turkish	president	nasa	bike	ftp	think	problem	pl	good	20
6	states	don	use	use	files	believe	know	145	time	11
7	israel	think	science	bit	information	don	card	1d9	ve	14
8	said	right	research	clipper	image	know	like	0t	people	17
9	children	public	build	number	send	just	using	34u	said	16
10	jews	make	section	phone	list	way	db	1t	year	13
11	000	state	center	like	use	like	scsi	3t	did	30
12	state	going	launch	cars	version	true	dos	giz	didn	24
13	new	privacy	time	just	server	question	disk	bhj	got	50
14	guns	private	high	engine	email	life	bit	wm	ll	18
15	israeli	security	earth	ground	pub	time	need	2di	going	19
16	vs	know	year	des	software	christian	pc	75u	way	27
17	military	new	rules	algorithm	cs	did	memory	2tm	game	40
18	years	rights	long	good	code	point	mac	cx	really	21
19	american	want	satellite	secret	window	bible	work	bxn	team	new