Assignment-4 : Data Analytics

Name : Ritesh sharma

Roll No : 191IT142

Kaggle Notebook Link : https://www.kaggle.com/riteshsharma12/191it142-it350-lab4

---

1. Find the clusters in the given dataset based on the content similarity and image similarity using k-means clustering and hierarchical clustering methods.

Importing necessary libraries

```
# importing necessary libraries
import numpy as np
import pandas as pd
import os
import pytesseract
import cv2
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
```

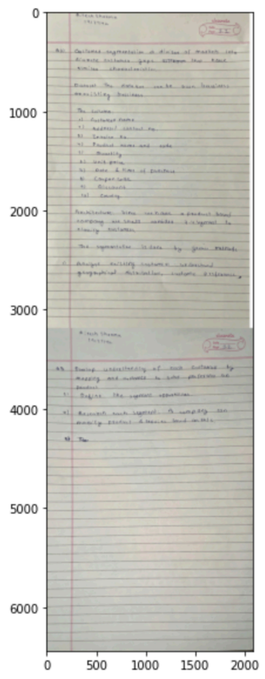
Storing images path in a list

[2]:
```
# storing images path in a list
path_name = []
for dirname, _, filenames in os.walk('../input/answer/Q3'):
    for filename in filenames:
        path_name.append(os.path.join(dirname, filename))
```

Printing sample datapoint from dataset

[3]:
```
img = mpimg.imread('../input/answer/Q3/191IT142.3.jpeg', cv2.IMREAD_GRAYSCALE)
plt.figure(figsize=(10,10))
plt.imshow(img, cmap ='gray')
```

[3]:
```
<matplotlib.image.AxesImage at 0x7efd024648d0>
```


0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
2000

Storing each image content in a list

```
# storing each image content in a list
file_content = {}
words = []
for i in range(0, len(path_name)):
    img = cv2.imread(path_name[i], cv2.IMREAD_GRAYSCALE)
    text = pytesseract.image_to_string(img)
    words.append(text)
    file_content[text] = path_name[i]
```

Method : **k-means clustering**

K-means clustering aims to partition data into k clusters in a way that data points in the same cluster are similar and data points in the different clusters are farther apart. Similarity of two points is determined by the distance between them. There are many methods to measure the distance.

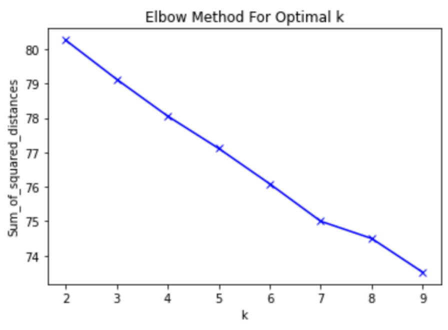
Elbow method for best k

The elbow method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center.

[6]:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

Sum_of_squared_distances = []
K = range(2,10)
for k in K:
    km = KMeans(n_clusters = k, max_iter = 200, n_init = 10)
    km = km.fit(X)
    Sum_of_squared_distances.append(km.inertia_)
plt.plot(K, Sum_of_squared_distances, 'bx-')
plt.xlabel('k')
plt.ylabel('Sum_of_squared_distances')
plt.title('Elbow Method For Optimal k')
plt.show()
```



Printing title and corresponding cluster number

[7]:
```
true_k = 6

title=[]
for i in file_content:
    title.append(file_content[i])

model = KMeans(n_clusters = true_k, init = 'k-means++', max_iter = 200, n_init = 10)
model.fit(X)
labels = model.labels_

wiki_cl = pd.DataFrame(list(zip(title,labels)),columns=['title','cluster'])
print(wiki_cl.sort_values(by = ['cluster']))
```

```
                                  title  cluster
79      ../input/answer/Q3/191IT136.3.jpeg        0
64  ../input/answer/Q3/191IT115.3.jpeg.jpg        0
76       ../input/answer/Q3/191IT146.3.jpg        0
13       ../input/answer/Q3/191IT102.3.png        0
62      ../input/answer/Q3/191IT249.3.jpeg        0
..                                    ...      ...
53      ../input/answer/Q3/191IT210.3.jpeg        5
15       ../input/answer/Q3/191IT158.3.jpg        5
70      ../input/answer/Q3/191IT208.3.jpeg        5
71      ../input/answer/Q3/191IT258.3.jpeg        5
10       ../input/answer/Q3/191IT135.3.png        5

[92 rows x 2 columns]
```

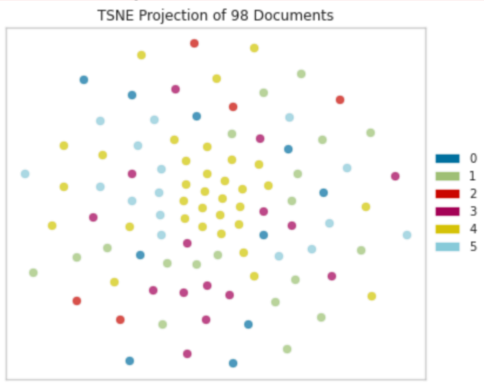
2. Plot t-SNE visualization for derived clusters.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised, non-linear technique primarily used for data exploration and visualizing high-dimensional data.

[9]:
```
import numpy as np
from sklearn.manifold import TSNE

from yellowbrick.text import TSNEVisualizer
tsne = TSNEVisualizer()
tsne.fit(X, labels)
tsne.show()
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/manifold/_t_sne.py:783: FutureWarning: The default initialization in TSNE will change
from 'random' to 'pca' in 1.2.
  FutureWarning,
/opt/conda/lib/python3.7/site-packages/sklearn/manifold/_t_sne.py:793: FutureWarning: The default learning rate in TSNE will change
from 200.0 to 'auto' in 1.2.
  FutureWarning,
```



[9]:
```
<AxesSubplot:title={'center':'TSNE Projection of 98 Documents'}>
```

3. Evaluate the clusters that are obtained using appropriate methods

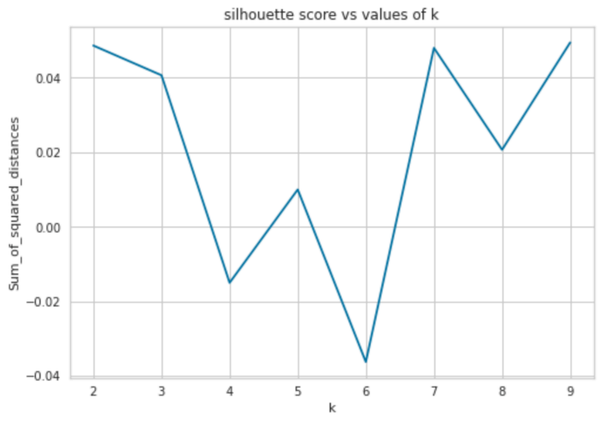
Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1. 1: Means clusters are well apart from each other and clearly distinguished.

```
from sklearn.metrics import silhouette_score
score=[]
c=[]

for k in K:
   km = KMeans(n_clusters=k, max_iter=200, n_init=10)
   labels = km.fit_predict(X)
   print(f'Silhouette Score(n={k}): {silhouette_score(X, labels)}')
   score.append(silhouette_score(X, labels))
   c.append(k)

plt.title('silhouette score vs values of k')
plt.plot(c, score, 'bx-')
plt.xlabel('k')
plt.ylabel('Sum_of_squared_distances')
plt.show()
```

```
Silhouette Score(n=2): 0.04859056381549375
Silhouette Score(n=3): 0.04062367478019349
Silhouette Score(n=4): -0.015015271806941123
Silhouette Score(n=5): 0.009973704975115355
Silhouette Score(n=6): -0.03624717641158025
Silhouette Score(n=7): 0.04795154848600708
Silhouette Score(n=8): 0.020655136934164587
Silhouette Score(n=9): 0.049411242400294576
```



1. Find the clusters in the given dataset based on the content similarity and image similarity using k-means clustering and hierarchical clustering methods.
Method : **Hierarchical clustering**

Printing title and corresponding cluster

```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=6, affinity='euclidean', linkage='ward')
labels_=cluster.fit_predict(X.toarray())
word_cloud1=pd.DataFrame(list(zip(title,labels_)),columns=['title','cluster'])
print(word_cloud1.sort_values(by=['cluster']))
```

```
                              title  cluster
0   ../input/answer/Q3/191IT204.3.jpeg        0
62  ../input/answer/Q3/191IT249.3.jpeg        0
61  ../input/answer/Q3/191IT253.3.jpeg        0
59   ../input/answer/Q3/191IT242.3.jpg        0
56   ../input/answer/Q3/191IT224.3.jpg        0
..                                  ...      ...
58  ../input/answer/Q3/191IT119.3.jpeg        5
47  ../input/answer/Q3/191IT131.3.jpeg        5
75   ../input/answer/Q3/191IT202.3.jpg        5
23  ../input/answer/Q3/191IT214.3.jpeg        5
81  ../input/answer/Q3/191IT128.3.jpeg        5

[92 rows x 2 columns]
```

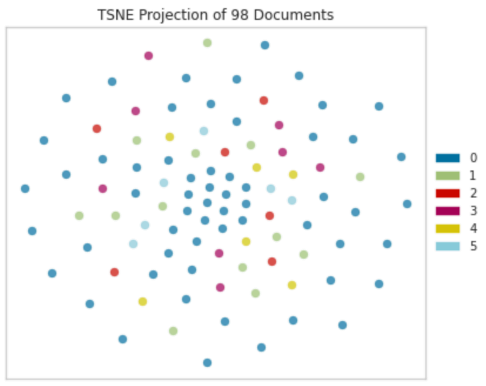
2. Plot t-SNE visualization for derived clusters.

[13]:

```
import numpy as np
from sklearn.manifold import TSNE

from yellowbrick.text import TSNEVisualizer
tsne = TSNEVisualizer()
tsne.fit(X, labels_)
tsne.show()
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/manifold/_t_sne.py:783: FutureWarning: The default initialization in TSNE will change
from 'random' to 'pca' in 1.2.
  FutureWarning,
/opt/conda/lib/python3.7/site-packages/sklearn/manifold/_t_sne.py:793: FutureWarning: The default learning rate in TSNE will change
from 200.0 to 'auto' in 1.2.
  FutureWarning,
```



[13…

```
<AxesSubplot:title={'center':'TSNE Projection of 98 Documents'}>
```

3. Evaluate the clusters that are obtained using appropriate methods

Method : **Silhouette Coefficient**

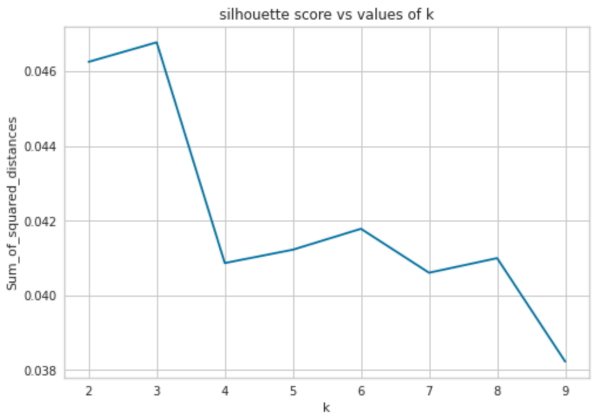
[15]:

```
from sklearn.metrics import silhouette_score
score=[]
c=[]

for k in K:
   km = AgglomerativeClustering(n_clusters=k, affinity='euclidean', linkage='ward')
   labels = km.fit_predict(X.toarray())
   print(f'Silhouette Score(n={k}): {silhouette_score(X, labels)}')
   score.append(silhouette_score(X, labels))
   c.append(k)

plt.title('silhouette score vs values of k')
plt.plot(c, score, 'bx-')
plt.xlabel('k')
plt.ylabel('Sum_of_squared_distances')
plt.show()
```

```
Silhouette Score(n=2): 0.04625511171242007
Silhouette Score(n=3): 0.046777694218318386
Silhouette Score(n=4): 0.04086325614179628
Silhouette Score(n=5): 0.04122147643430726
Silhouette Score(n=6): 0.04178163009931429
Silhouette Score(n=7): 0.04060404566897715
Silhouette Score(n=8): 0.040994778846469
Silhouette Score(n=9): 0.03822304470436569
```


silhouette score vs values of k
Sum_of_squared_distances
k