Assignment_3

November 26, 2023

6375 ML - Assignment 3

Aaryan Singh - axc230019

Nikunj Gohil - ndg220000

---

1. Importing libraries

[5]:
```
import numpy as np
import re
import pandas as pd
from tabulate import tabulate
```

2. Importing dataset and preprocessing

[6]:
```
file_path = 'https://raw.githubusercontent.com/aaryans99/
 ↪CS-6375-Machine-Learning/main/Assignment%203/Health-Tweets/usnewshealth.txt'␣
 ↪ # Replace with the path to your text file
# Define column names
column_names = ['tweet id', 'date and time', 'tweet']

# Read the text file with specified column names and selecting the specified␣
 ↪column
df = pd.read_csv(file_path, sep='|', names=column_names, usecols=['tweet'])


# Display the DataFrame with the added column headings
df
```

[6]:

	tweet
0	Planning to hire a personal trainer? Read thes…
1	RT @AnnaMedaris: Any dads out their who strugg…
2	America's problem with diabetes in one map: ht…
3	Think water & fiber will cure your constip…
4	About to lose it? Here, try one of these offic…
…	…
1390	RT @AnnaMedaris: Have you tried a #dance party…
1391	Going gray early? Here's how to stop it. http:…

```
1392  Sure, we all get nervous sometimes. But how to…
1393  RT @leonardkl: Millions have signed up for hea…
1394  RT @leonardkl: Are you getting #healthinsuranc…

[1395 rows x 1 columns]
```

[7]:
```
def clean_text(text):
    # Remove words starting with @ symbol
    text = re.sub(r'@\w+\s?', '', text)

    # Remove hashtag symbols and convert URLs to an empty string
    text = re.sub(r'#', '', text)
    text = re.sub(r'http\S+|www\S+', '', text)

    # Convert text to lowercase
    text = text.lower()

    return text

df['tweet'] = df['tweet'].apply(clean_text)
df
```

[7]:
```
                                                tweet
0     planning to hire a personal trainer? read thes…
1     rt : any dads out their who struggled w/ depre…
2     america's problem with diabetes in one map:   by
3     think water &amp;amp; fiber will cure your constip…
4     about to lose it? here, try one of these offic…
…                                                   …
1390  rt : have you tried a dance party fitness clas…
1391           going gray early? here's how to stop it.
1392  sure, we all get nervous sometimes. but how to…
1393  rt : millions have signed up for health insura…
1394  rt : are you getting healthinsurance for the f…

[1395 rows x 1 columns]
```

3. Performing k-means clustering

[9]:
```
# Function to calculate Jaccard distance between two sets
def jaccard_distance(set1, set2):
    intersection = len(set1.intersection(set2))
    union = len(set1.union(set2))
    return 1 - (intersection / union) if union != 0 else 0  # Avoid division by␣
 ↪zero

# Function to perform K-means clustering with Jaccard distance
def kmeans_clustering_jaccard(k, data):
```

```
    centroids = data['tweet'].iloc[:k].apply(lambda x: set(x.lower().split()))

    tweet_sets = data['tweet'].apply(lambda x: set(x.lower().split()))

    clusters = [[] for _ in range(k)]
    for tweet_set in tweet_sets:
        distances = [jaccard_distance(tweet_set, centroid) for centroid in␣
 ↪centroids]
        closest_centroid_index = distances.index(min(distances))
        clusters[closest_centroid_index].append(tweet_set)

    # Calculate SSE (sum of Jaccard distances)
    sse = 0
    for i, centroid in enumerate(centroids):
        cluster_sets = clusters[i]
        sse += sum(jaccard_distance(tweet_set, centroid) ** 2 for tweet_set in␣
 ↪cluster_sets)

    return sse, clusters

# Perform K-means clustering for different values of K
results = []
for k in range(2, 12):
    sse, clusters = kmeans_clustering_jaccard(k, df)
    cluster_sizes = [len(cluster) for cluster in clusters]
    results.append((k, sse, cluster_sizes))

# Display the results in tabular format using tabulate
table_headers = ["Value of K", "SSE", "Size of each cluster"]
table_data = [[result[0], result[1], result[2]] for result in results]

print(tabulate(table_data, headers=table_headers, tablefmt="grid"))
```

```
+--------------+---------+-----------------------------------------------------+
|   Value of K |     SSE | Size of each cluster                                |
+==============+=========+=====================================================+
|            2 | 1279.98 | [1023, 372]                                         |
+--------------+---------+-----------------------------------------------------+
|            3 | 1262.56 | [783, 343, 269]                                     |
+--------------+---------+-----------------------------------------------------+
|            4 | 1248.86 | [671, 314, 261, 149]                                |
+--------------+---------+-----------------------------------------------------+
|            5 | 1236.67 | [564, 287, 235, 149, 160]                           |
+--------------+---------+-----------------------------------------------------+
|            6 | 1218.06 | [416, 261, 217, 62, 142, 297]                       |
+--------------+---------+-----------------------------------------------------+
|            7 | 1209.34 | [314, 255, 207, 55, 134, 288, 142]                  |
```

8	1193.34	[238, 228, 178, 43, 99, 273, 132, 204]
9	1189.05	[236, 222, 156, 43, 97, 269, 132, 64, 176]
10	1183.87	[222, 211, 134, 43, 80, 262, 132, 64, 176, 71]
11	1168.01	[185, 176, 113, 40, 71, 203, 96, 59, 167, 67, 218]