Step 0 - install and import dependencies

In [1]:
```
!pip install pythainlp
!pip install tensorflow_text
!pip install umap-learn
```

```
Collecting pythainlp
  Downloading pythainlp-2.3.2-py3-none-any.whl (11.0 MB)
     |████████████████████████████████| 11.0 MB 4.5 MB/s
Collecting python-crfsuite>=0.9.6
  Downloading python_crfsuite-0.9.7-cp37-cp37m-manylinux1_x86_64.whl (743 kB)
     |████████████████████████████████| 743 kB 58.4 MB/s
Collecting tinydb>=3.0
  Downloading tinydb-4.5.2-py3-none-any.whl (23 kB)
Requirement already satisfied: requests>=2.22.0 in /usr/local/lib/python3.7/dist-packages
(from pythainlp) (2.23.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/p
ython3.7/dist-packages (from requests>=2.22.0->pythainlp) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-package
s (from requests>=2.22.0->pythainlp) (2021.10.8)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages
(from requests>=2.22.0->pythainlp) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (fro
m requests>=2.22.0->pythainlp) (2.10)
Requirement already satisfied: typing-extensions<4.0.0,>=3.10.0 in /usr/local/lib/python3.
7/dist-packages (from tinydb>=3.0->pythainlp) (3.10.0.2)
Installing collected packages: tinydb, python-crfsuite, pythainlp
Successfully installed pythainlp-2.3.2 python-crfsuite-0.9.7 tinydb-4.5.2
Collecting tensorflow_text
  Downloading tensorflow_text-2.7.3-cp37-cp37m-manylinux2010_x86_64.whl (4.9 MB)
     |████████████████████████████████| 4.9 MB 5.1 MB/s
Requirement already satisfied: tensorflow-hub>=0.8.0 in /usr/local/lib/python3.7/dist-pack
ages (from tensorflow_text) (0.12.0)
Requirement already satisfied: tensorflow<2.8,>=2.7.0 in /usr/local/lib/python3.7/dist-pac
kages (from tensorflow_text) (2.7.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from
tensorflow<2.8,>=2.7.0->tensorflow_text) (3.1.0)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /usr/local/lib/python3.7/dist-pac
kages (from tensorflow<2.8,>=2.7.0->tensorflow_text) (2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from
tensorflow<2.8,>=2.7.0->tensorflow_text) (1.15.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-p
ackages (from tensorflow<2.8,>=2.7.0->tensorflow_text) (3.10.0.2)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /usr/local/lib/python3.7/dist-package
s (from tensorflow<2.8,>=2.7.0->tensorflow_text) (0.37.0)
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (12.0.0)
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages (f
rom tensorflow<2.8,>=2.7.0->tensorflow_text) (0.12.0)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in /usr/local/lib/python3.7/dist-packa
ges (from tensorflow<2.8,>=2.7.0->tensorflow_text) (2.7.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (fr
om tensorflow<2.8,>=2.7.0->tensorflow_text) (1.13.3)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (3.3.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (3.17.3)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (1.1.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist
-packages (from tensorflow<2.8,>=2.7.0->tensorflow_text) (1.1.2)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packag
es (from tensorflow<2.8,>=2.7.0->tensorflow_text) (0.2.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages
```

```
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (1.6.3)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0rc0 in /usr/local/lib/pytho
n3.7/dist-packages (from tensorflow<2.8,>=2.7.0->tensorflow_text) (2.7.0)
Requirement already satisfied: tensorboard~=2.6 in /usr/local/lib/python3.7/dist-packages
(from tensorflow<2.8,>=2.7.0->tensorflow_text) (2.7.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packag
es (from tensorflow<2.8,>=2.7.0->tensorflow_text) (1.42.0)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in /usr/local/lib/python3.7/dist-package
s (from tensorflow<2.8,>=2.7.0->tensorflow_text) (0.4.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /usr/local/lib/pyth
on3.7/dist-packages (from tensorflow<2.8,>=2.7.0->tensorflow_text) (0.22.0)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (fr
om tensorflow<2.8,>=2.7.0->tensorflow_text) (1.19.5)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages
(from h5py>=2.9.0->tensorflow<2.8,>=2.7.0->tensorflow_text) (1.5.2)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.
7/dist-packages (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (0.4.6)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packag
es (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (2.23.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (0.
6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/d
ist-packages (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (1.8.0)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages
(from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages
(from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (3.3.6)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-package
s (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (57.4.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-pack
ages (from tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (1.35.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-pack
ages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_tex
t) (0.2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-pac
kages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_te
xt) (4.2.4)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (fr
om google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (4.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-p
ackages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->
tensorflow_text) (1.3.0)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-pa
ckages (from markdown>=2.6.8->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text)
(4.8.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from i
mportlib-metadata>=4.4->markdown>=2.6.8->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorf
low_text) (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packa
ges (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow<2.8,>
=2.7.0->tensorflow_text) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-package
s (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (2
021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/p
ython3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow<2.8,>=2.7.0
->tensorflow_text) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages
(from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (3.
0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (fro
m requests<3,>=2.21.0->tensorboard~=2.6->tensorflow<2.8,>=2.7.0->tensorflow_text) (2.10)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages
(from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensor
flow<2.8,>=2.7.0->tensorflow_text) (3.1.1)
```

```
Installing collected packages: tensorflow-text
Successfully installed tensorflow-text-2.7.3
Collecting umap-learn
  Downloading umap-learn-0.5.2.tar.gz (86 kB)
     |████████████████████████████████| 86 kB 2.9 MB/s
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from
umap-learn) (1.19.5)
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.7/dist-package
s (from umap-learn) (1.0.1)
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-packages (from
umap-learn) (1.4.1)
Requirement already satisfied: numba>=0.49 in /usr/local/lib/python3.7/dist-packages (from
umap-learn) (0.51.2)
Collecting pynndescent>=0.5
  Downloading pynndescent-0.5.5.tar.gz (1.1 MB)
     |████████████████████████████████| 1.1 MB 40.3 MB/s
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from umap-l
earn) (4.62.3)
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in /usr/local/lib/python3.7/dis
t-packages (from numba>=0.49->umap-learn) (0.34.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from
numba>=0.49->umap-learn) (57.4.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (fro
m pynndescent>=0.5->umap-learn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packa
ges (from scikit-learn>=0.22->umap-learn) (3.0.0)
Building wheels for collected packages: umap-learn, pynndescent
  Building wheel for umap-learn (setup.py) ... done
  Created wheel for umap-learn: filename=umap_learn-0.5.2-py3-none-any.whl size=82709 sha2
56=51b6f1fd7ed2271e89c6268fe52feba14726a7d2a521f115892f977360a29eb2
  Stored in directory: /root/.cache/pip/wheels/84/1b/c6/aaf68a748122632967cef4dffef68224eb
16798b6793257d82
  Building wheel for pynndescent (setup.py) ... done
  Created wheel for pynndescent: filename=pynndescent-0.5.5-py3-none-any.whl size=52603 sh
a256=59ad7be091d6edce0786f628045637e742753f4cca5574b8e197cc67cc1c580a
  Stored in directory: /root/.cache/pip/wheels/af/e9/33/04db1436df0757c42fda8ea6796d7a8586
e23c85fac355f476
Successfully built umap-learn pynndescent
Installing collected packages: pynndescent, umap-learn
Successfully installed pynndescent-0.5.5 umap-learn-0.5.2
```

In [2]:

```
import numpy as np
import pandas as pd
import re

import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_text
import umap

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

from sklearn.cluster import AgglomerativeClustering
from sklearn.neighbors import kneighbors_graph

import pythainlp
from pythainlp.corpus.common import thai_words
from pythainlp.util import Trie
import collections
```

In [3]:

```
module_url = 'https://tfhub.dev/google/universal-sentence-encoder-multilingual/3' #'https.

model = hub.load(module_url)
```

In [4]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [10]:
```
df = pd.read_csv("drive/MyDrive/Colab Notebooks/CRM - Voice of Customers/Wongnai Reviews -
```

In [ ]:
```
df.head()
```

Out[ ]:

	Review ID	Review
0	1	เป็นคนที่ชอบทาน Macchiato เป็นประจำ มีวันนึงเด...
1	2	Art of Coffee Kasetsart เป็นร้านกาแฟรสชาติเยี่...
2	3	กวงทะเลเผา อาหารทะเลเค้าสดจริงๆเนื้อปูหวานไม่ค...
3	4	วันนี้มีโอกาสตื่นเช้าครับเลยถึงโอกาสออกมาหาอะไ...
4	5	ชอบมาทานร้านนี้ถ้าอยากกินอาหารเวียดนามใกล้บ้าน...

Step 1 - document embedding and dimension reduction

In [11]:
```
#embed sentences using Universal Sentence Encoder (USE)

embed_comments_array = model(df['Review'].values).numpy()
embed_comments_array
```

Out[11]:
```
array([[ 0.08993827,  0.01941084,  0.03787038, ..., -0.03488849,
         0.06299512,  0.04635989],
       [ 0.00634244,  0.00814594,  0.03071941, ..., -0.01478723,
        -0.03080936, -0.03316405],
       [ 0.0633687 , -0.02027139, -0.05077003, ..., -0.06530775,
        -0.00952999, -0.03439987],
       ...,
       [ 0.08775924,  0.03609736,  0.01263062, ..., -0.03102781,
        -0.03361677,  0.01928871],
       [ 0.05691195,  0.05381691, -0.0399575 , ..., -0.06598807,
        -0.05390478, -0.01037725],
       [ 0.0777048 ,  0.05080631,  0.02680681, ..., -0.0061413 ,
        -0.01313567,  0.02236264]], dtype=float32)
```

In [29]:
```
#reduce array dimensions using umap (you can chagne n_components)

reducer = umap.UMAP(random_state=42,n_components=100,n_neighbors=50,min_dist=0.5)
umap_embed_comments_array = reducer.fit_transform(embed_comments_array)
```

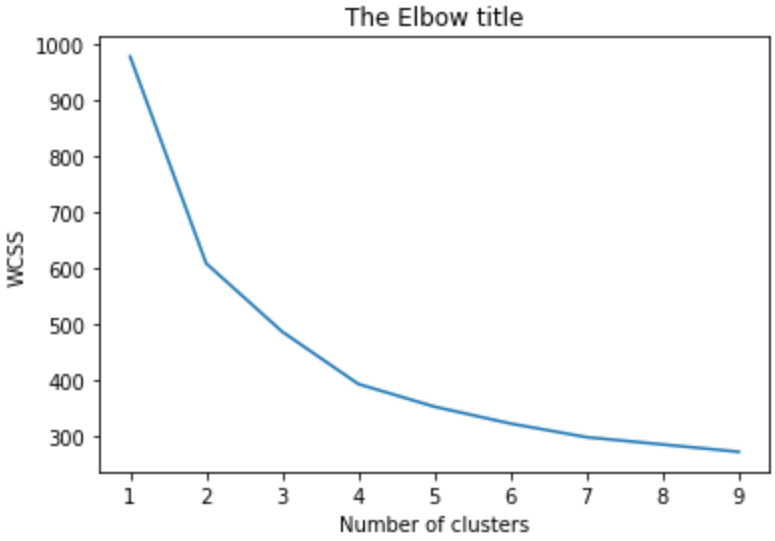
Step 2 - document clustering using KMeans

In [30]:
```
#run kmeans with various number of k. evaluate no. of k based on the elbow plot

wcss=[]
max_k = 10
for i in range(1, max_k):
  kmeans = KMeans(i)
  kmeans.fit(umap_embed_comments_array)
  wcss_iter = kmeans.inertia_
  wcss.append(wcss_iter)
```

```
number_clusters = range(1, max_k)
plt.plot(number_clusters,wcss)
plt.title('The Elbow title')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
```

Out[30]:
```
Text(0, 0.5, 'WCSS')
```



In [69]:
```
#run kmeans with no. of clusters you see fit the most

k = 4

kmeans = KMeans(n_clusters = k)
kmeans.fit(umap_embed_comments_array)

df['KMeans ID'] = kmeans.labels_
```

In [70]:
```
#merge all reviews of each cluster into one big sentence

df_kmeans = pd.DataFrame(columns=["KMeans ID", "texts"])


for i in range(0, k):
  row = []
  row.append(i)
  row.append(df['Review'][df['KMeans ID'] == i].to_string())
  df_kmeans.loc[len(df_kmeans)] = row
```

In [71]:
```
df_kmeans
```

Out[71]:

	KMeans ID	texts
0	0	2 กวงทะเลเผา อาหารทะเลเค้าสดจริงๆเนื้อปูห...
1	1	13 เคยเป็นไหมกันไหมคะ หลังอาหารมื้อใหญ่ ต่...
2	2	3 วันนี้มีโอกาสตื่นเช้าครับเลยถึงโอกาสออก...
3	3	0 เป็นคนที่ชอบทาน Macchiato เป็นประจำ มีว...

In [72]:
```
#create regex compiler for removal of a character you don't want
```

```
special_characters = "/[!@#$%^&*']/g"

specialchar_pattern = re.compile(special_characters)
```

In [73]:
```
#create regex compiler for removal of any emoji

emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           "]+", flags=re.UNICODE)
```

In [74]:
```
#create regex compiler for removal of digit

number_pattern = re.compile("[0-9]")
```

In [75]:
```
#create regex compiler for removal of white space

space_pattern = re.compile("\s+")
```

In [76]:
```
#create regex compiler for removal of .

dot_pattern = re.compile(r"\.+")
```

In [77]:
```
#create regex compiler for removal of \

backslash_pattern = re.compile(r"\\+")
```

In [78]:
```
#Remove web pattern
web1_pattern = re.compile(r"[^\s]*.com[^\s]*")
web2_pattern = re.compile(r"[^\s]*www.[^\s]*")
```

In [79]:
```
#Remove star word
star_pattern = re.compile(r"[^\s]*[\*]+[^\s]*")
```

In [139…
```
#define a function to tokenize a sentence into words - you can define words you want to re

stopwords = list(pythainlp.corpus.thai_stopwords())
removed_words = ['u', 'b', 'n', 'nn', 'nn-', '\n', 'ร้าน', 'ดิชั้น', 'แย่มาก','บอ','กก','น','ร้า'
,'ๆๆๆ','ดีแต่','แก้','เฉพาะเรื่อง','ม่ะ','จ้ย','ๆเริศ','มากเกินไป','โถม','ย้ำ']
screening_words = stopwords + removed_words

new_words = {"สตารบัก"}

words = new_words.union(thai_words())

custom_dictionary_trie = Trie(words)

def tokenize_to_list(sentence):
  merged = []
  words = pythainlp.word_tokenize(str(sentence), engine='newmm', custom_dict=custom_dictio
  for word in words:
    if word not in screening_words:
```

```
        merged.append(word)
    return merged
```

In [140…

```
#clean and tokenize sentences. count the occurences of each word

df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: emoji_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: specialchar_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: number_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: space_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: dot_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: backslash_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: web1_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: web2_pattern.sub(r'', x))
df_kmeans['texts'] = df_kmeans['texts'].apply(lambda x: star_pattern.sub(r'', x))
df_kmeans['texts_tokenized'] = df_kmeans['texts'].apply(lambda x: tokenize_to_list(x))
df_kmeans['texts_count'] = df_kmeans['texts_tokenized'].apply(lambda x: collections.Counte
```

In [141…

```
#results of tokenization

df_kmeans
```

Out[141…

	KMeans ID	texts	texts_tokenized	texts_count
0	0		[]	[]
1	1	เคยเป็นไหมกันไหมคะหลังอาหารมื้อใหญ่ต่อให้อิ่ระ...	[ไหม, ไหม, หลังอาหาร, มื้อ, ต่อให้, อิ่, มุ่งห...	[(ชา, 18), (ไข่มุก, 14), (นม, 13), (ทาน, 6), (...
2	2	วันนี้มีโอกาสตื่นเช้าครับเลยถึงโอกาสออกมาหาอะไ...	[มีโอกาส, ตื่น, เช้า, โอกาส, มาหา, อะ, ไช, อบ,...	[(กิน, 12), (ทาน, 8), (อร่อย, 8), (อาหาร, 6), ...
3	3	เป็นคนที่ชอบทานMacchiatoเป็นประจำมีวันนึงเดArt...	[คน, ชอบ, ทาน, Macchiato, เป็นประจำ, นึง, เด, ...	[(ร้านกาแฟ, 24), (กาแฟ, 20), (ทาน, 9), (ชอบ, 6...

In [142…

```
#show top keywords of each cluster

top_N_words = 10

for i in range(0, len(df_kmeans)):
  print(f"Cluster ID : {i}\n")
  print(f"Most common words include : {list(df_kmeans['texts_count'][i])[:top_N_words]}\n"
```

```
Cluster ID : 0

Most common words include : []

Cluster ID : 1

Most common words include : [('ชา', 18), ('ไข่มุก', 14), ('นม', 13), ('ทาน', 6), ('เครื่องดื่ม', 4), ('รีวิว', 4), ('ตั้งอยู่', 3), ('ลอง', 3), ('เดิน', 3), ('ไต้หวัน', 3)]

Cluster ID : 2

Most common words include : [('กิน', 12), ('ทาน', 8), ('อร่อย', 8), ('อาหาร', 6), ('ร้านอาหาร', 5), ('กาแฟ', 5), ('พาย', 4), ('ซื้อ', 4), ('ชอบ', 4), ('รีวิว', 4)]

Cluster ID : 3

Most common words include : [('ร้านกาแฟ', 24), ('กาแฟ', 20), ('ทาน', 9), ('ชอบ', 6), ('กิน',
```

6), ('น่ารัก', 5), ('นั่ง', 5), ('เจอ', 5), ('บรรยากาศ', 5), ('คน', 4)]

Step 3 - document clustering using Agglomorative Clustering with cosine similarity

In [143…

```
#clustering using agglomorative clustering

knn_graph = kneighbors_graph(embed_comments_array, 5, include_self=False)
model = AgglomerativeClustering(linkage="average", connectivity=knn_graph, n_clusters=4, a
model.fit(embed_comments_array)
df['Agglomerative ID'] = model.labels_
```

In [144…

```
#merge all reviews of each cluster into one big sentence

df_Agglomerative = pd.DataFrame(columns=["Agglomerative ID", "texts"])


for i in range(0, k):
  row = []
  row.append(i)
  row.append(str(df['Review'][df['Agglomerative ID'] == i].tolist()))
  df_Agglomerative.loc[len(df_Agglomerative)] = row
```

In [145…

```
#clean and tokenize sentences. count the occurences of each word

df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: emoji_pattern.sub(r'
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: specialchar_pattern.
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: number_pattern.sub(r
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: space_pattern.sub(r'
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: dot_pattern.sub(r'',
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: backslash_pattern.su
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: web1_pattern.sub(r''
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: web2_pattern.sub(r''
df_Agglomerative['texts'] = df_Agglomerative['texts'].apply(lambda x: star_pattern.sub(r''
df_Agglomerative['texts_tokenized'] = df_Agglomerative['texts'].apply(lambda x: tokenize_t
df_Agglomerative['texts_count'] = df_Agglomerative['texts_tokenized'].apply(lambda x: coll
```

In [146…

```
#show top keywords of each cluster

top_N_words = 10

for i in range(0, len(df_Agglomerative)):
  print(f"Cluster ID : {i}\n")
  print(f"Most common words include : {list(df_Agglomerative['texts_count'][i])[:top_N_wor
```

```
Cluster ID : 0

Most common words include : []

Cluster ID : 1

Most common words include : [('แดงโม', 22), ('น้ำ', 8), ('ปั่น', 6), ('เนื้อ', 6), ('เลือก', 4),
('ซื้อ', 4), ('ดื่ม', 4), ('พันธุ์', 3), ('รับประทาน', 3), ('อาหาร', 3)]

Cluster ID : 2

Most common words include : [('ปัง', 4), ('ภูเขาไฟ', 3), ("['", 1), ('ร้อน', 1), ('เข้ากัน', 1),
('หวาน', 1), ('อร่อย', 1), ('กาแฟ', 1), ('เย็น', 1), ('เมนู', 1)]

Cluster ID : 3
```

```
Most common words include : [('นม', 3), ('แน่น', 2), ('tamp', 2), ('เท', 2), ("['", 1), ('ข
นม', 1), ('review', 1), ('กาแฟร้อน', 1), ('nTamp', 1), ('-', 1)]
```

Step 4 - result discussion

จาก K-means clustering จะได้กลุ่มลูกค้า 3 กลุ่ม

กลุ่มที่ 1 คือลูกค้าที่ชอบทานชานมไข่มุก

กลุ่มที่ 2 คือลูกค้าที่ชอบทานที่ร้านอาหาร

กลุ่มที่ 3 คือลูกค้าที่ชอบเข้าร้านกาแฟเพื่อดื่มด่ำบรรยากาศและซื้อกาแฟ

จาก agglomorative clustering จะได้กลุ่มลูกค้า 3 กลุ่ม

กลุ่มที่ 1 คือลูกค้าที่ชอบดื่มน้ำแดงโมปั่นและรับประทานเนื้อ

กลุ่มที่ 2 คือลูกค้าที่ชอบทานปังภูเขาไฟพร้อมกับกาแฟร้อนหรือกาแฟเย็น (ลูกค้าชอบของหวาน)

กลุ่มที่ 3 คือลูกค้าที่ชอบทานขนมปังนมพร้อมกับกาแฟร้อน