In [ ]:

```
#!pip install vaderSentiment
#!pip uninstall scikit-learn -y
#!pip install -U scikit-learn
#!pip install gensim

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from google.colab import drive
from collections import Counter

import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from gensim import corpora
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
nltk.download('stopwords')
nltk.download('punkt')

from sklearn.cluster import KMeans
from scipy.spatial.distance import pdist, squareform
from scipy.cluster.hierarchy import linkage, dendrogram
from scipy.cluster.hierarchy import fcluster
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors
from sklearn.metrics import silhouette_score

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree, export_text
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics import RocCurveDisplay
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier, RandomForestClassifier
from sklearn.model_selection import ValidationCurveDisplay, validation_curve
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [ ]:

```
drive.mount('/content/drive')
df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/INFO3237_Spring2024/HW/Health_Data.csv', na_values = ["N/A"])
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

In [ ]:

```
# Dataset size
df.shape
```

Out[ ]:

```
(3460, 10)
```

In [ ]:

```
# Drop missing values
df.dropna(inplace=True)
df.shape
```

Out[ ]:

```
(3452, 10)
```

In [ ]:

```
# Summary statistics
df.describe()
df.dtypes
```

Out[ ]:

```
Reddit_ID               object
Subreddits              object
Titles                  object
Body                    object
Author                  object
Initial Create          object
Retrieved_date          object
Date_of_collection      object
Number of Comments     float64
Decision               float64
dtype: object
```

In [ ]:

```
df.head()
```

Out[ ]:

	Reddit_ID	Subreddits	Titles	Body	Author	Initial Create	Retrieved_date	Date_of_collection	Number of Comments	Decision
0	xn3aad	loseit	Why am I gaining weight on a healthy diet?	Hello, \nI really need some advice for someon...	Shinigami_Sadies	9/24/2022 16:33	9/24/2022 16:33	2022_9_23	29.0	0.0
1	xyz6l9	xxfitness	I ran a sub 30 5K	I know that’s easy for most people but I haven...	Choice_Ad522	10/8/2022 13:59	10/8/2022 13:59	2022_10_7	3.0	1.0
2	x6u9cp	loseit	I am losing my mind instead of weight &gt;:(	I am currently 276, so 79 pounds down from my ...	peppermintwhitemocha	9/5/2022 19:09	9/5/2022 19:09	2022_9_4	22.0	0.0
3	ybse4o	bodybuilding	Please answer!!!	Do men bodybuilders use waist trainers?????	GrowthMobile	10/23/2022 16:41	10/23/2022 16:42	2022_10_22	12.0	1.0
4	x0yt2o	loseit	Worried about muscle loss	Hey\nSo basically I’m a 30 year old male Im 6f...	Takedownkd	8/29/2022 16:22	8/29/2022 16:22	2022_8_28	4.0	0.0

In [ ]:

```
state = df.groupby('Subreddits')[['Number of Comments']].agg(['sum', 'mean', 'count']).reset_index()
sorted_state = state.sort_values(by=('Number of Comments', 'mean'), ascending=False)
print(sorted_state)
```

```
           Subreddits Number of Comments                 
                                     sum       mean count
0  EatCheapAndHealthy             4909.0  24.918782   197
8               vegan             6430.0  19.484848   330
7        orangetheory             6551.0  18.610795   352
2   bodyweightfitness             5107.0  17.794425   287
5        nattyorjuice              546.0  14.368421    38
3            crossfit             2112.0  14.174497   149
6           nutrition             4710.0  12.975207   363
4              loseit            12326.0  11.595484  1063
9           xxfitness             4491.0   9.980000   450
1        bodybuilding             2194.0   9.838565   223
```

In [ ]:

```
# Merge two columns
df['Titles_Body'] = df['Titles']+' '+df['Body']
df.head()
```

Out[ ]:

	Reddit_ID	Subreddits	Titles	Body	Author	Initial Create	Retrieved_date	Date_of_collection	Number of Comments	Decision	Titles_Body
0	xn3aad	loseit	Why am I gaining weight on a healthy diet?	Hello, \nI really need some advice for someon...	Shinigami_Sadies	9/24/2022 16:33	9/24/2022 16:33	2022_9_23	29.0	0.0	Why am I gaining weight on a healthy diet? Hel...
1	xyz6l9	xxfitness	I ran a sub 30 5K	I know that’s easy for most people but I haven...	Choice_Ad522	10/8/2022 13:59	10/8/2022 13:59	2022_10_7	3.0	1.0	I ran a sub 30 5K I know that’s easy for most ...
2	x6u9cp	loseit	I am losing my mind instead of weight &gt;:(	I am currently 276, so 79 pounds down from my ...	peppermintwhitemocha	9/5/2022 19:09	9/5/2022 19:09	2022_9_4	22.0	0.0	I am losing my mind instead of weight &gt;:(I...
3	ybse4o	bodybuilding	Please answer!!!	Do men bodybuilders use waist trainers?????	GrowthMobile	10/23/2022 16:41	10/23/2022 16:42	2022_10_22	12.0	1.0	Please answer!!! Do men bodybuilders use waist...
4	x0yt2o	loseit	Worried about muscle loss	Hey\nSo basically I’m a 30 year old male Im 6f...	Takedownkd	8/29/2022 16:22	8/29/2022 16:22	2022_8_28	4.0	0.0	Worried about muscle loss Hey\nSo basically I’...

In [ ]:

```python
# Customize your pre-processing
def preprocess_text(text):
    # remove Punctuation
    text = re.sub(r'[^\w\s]', '', text)
    # tokenize
    tokens = word_tokenize(text)
    # convert to lowercase
    tokens = [t.lower() for t in tokens]
    # remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [t for t in tokens if t not in stop_words]
    # stemming
    stemmer = PorterStemmer()
    tokens = [stemmer.stem(t) for t in tokens]
    return tokens
```

In [ ]:

```
# Pre-process post content
df['processed'] = df['Titles_Body'].apply(preprocess_text)
```

In [ ]:

```
# Double check pre-processed posts
print(df['processed'][:10])
```

```
0    [gain, weight, healthi, diet, hello, realli, n...
1    [ran, sub, 30, 5k, know, that, easi, peopl, ha...
2    [lose, mind, instead, weight, gt, current, 276...
3    [pleas, answer, men, bodybuild, use, waist, tr...
4    [worri, muscl, loss, hey, basic, im, 30, year,...
5    [mount, pull, bar, miss, screw, recent, bought...
6    [5, week, alreadi, see, differ, small, mention...
7    [healthi, cocacola, make, skinni, fat, eat, lo...
8          [sick, cant, go, gym, sad, h, cope, happen]
9    [lose, weight, even, massiv, calori, deficit, ...
Name: processed, dtype: object
```

In [ ]:

```
# Count Word frequency
word_counts = Counter()
for word in df['processed']:
  word_counts.update(word)
word_counts
```

```
Out[ ]: Counter({'gain': 778,
                 'weight': 3058,
                 'healthi': 528,
                 'diet': 786,
                 'hello': 178,
                 'realli': 1124,
                 'need': 930,
                 'advic': 533,
                 'someon': 268,
                 'pco': 25,
                 'loos': 138,
                 'last': 564,
                 'year': 1153,
                 'final': 188,
                 'reach': 161,
                 'goal': 587,
                 'ideal': 60,
                 'howev': 250,
                 'eat': 2316,
                 'littl': 456,
                 'food': 997,
                 'feint': 2,
                 'lot': 716,
                 'time': 1636,
                 'life': 434,
                 'end': 390,
                 'bmi': 79,
                 'doctor': 120,
                 'said': 214,
                 'goodbut': 1,
                 'knew': 53,
                 'wasnt': 150,
                 'decid': 207,
                 'better': 408,
                 'go': 1620,
                 'focus': 93,
                 'nutrit': 151,
                 'right': 466,
                 'type': 120,
                 'rather': 89,
                 'less': 274,
                 'even': 827,
                 'though': 293,
                 'good': 836,
                 'nolong': 1,
                 'rapidli': 9,
                 'notic': 219,
                 'peopl': 640,
                 'care': 136,
                 'due': 219,
                 'health': 301,
                 'risk': 29,
                 'warn': 17,
                 'activ': 262,
                 '8': 249,
                 'hour': 325,
                 'work': 1475,
                 'feet': 46,
                 'break': 134,
                 '5': 552,
                 'day': 2436,
                 'week': 1720,
                 'workday': 1,
                 '4am': 2,
                 'vegetablelentil': 1,
                 'soup': 49,
                 '1': 862,
                 'tin': 5,
                 '8am': 6,
                 'hand': 135,
                 'grape': 13,
                 '2': 898,
                 'rice': 163,
                 'cake': 49,
                 'low': 307,
                 'fat': 859,
                 '12': 337,
                 'small': 211,
                 'wholegrain': 4,
                 'bread': 112,
                 'roll': 43,
                 'beetroot': 3,
                 'salad': 78,
                 'tomato': 54,
                 'butter': 75,
                 'spread': 17,
                 '2pm': 2,
                 'eggcup': 1,
                 'size': 174,
                 'amount': 229,
                 'nut': 44,
                 '4pm': 1,
                 'stirfri': 4,
                 'veg': 32,
                 'bean': 67,
                 'teaspoon': 17,
                 'cook': 309,
                 'weekend': 121,
                 'usual': 324,
                 'dinner': 185,
                 'lunch': 153,
                 'vari': 30,
                 'breakfast': 118,
                 'toast': 18,
                 'sometim': 209,
                 'snack': 187,
                 'crisp': 7,
                 'drink': 309,
                 'tea': 52,
                 'coffe': 51,
                 'water': 334,
                 'help': 924,
                 'chang': 453,
                 'vegetarian': 95,
                 'dairi': 54,
                 'give': 353,
                 'acn': 14,
                 'tri': 1369,
                 'consum': 132,
                 'possibl': 235,
                 'keep': 496,
                 'like': 2257,
                 'fear': 31,
                 'effect': 115,
                 'mental': 152,
                 'poorli': 6,
                 'ran': 29,
                 'sub': 89,
                 '30': 602,
                 '5k': 15,
                 'know': 1360,
                 'that': 378,
                 'easi': 158,
                 'havent': 235,
                 'abl': 314,
                 'run': 480,
                 '56': 69,
                 'lose': 1380,
                 'mind': 152,
                 'instead': 152,
                 'gt': 64,
                 'current': 401,
                 '276': 4,
                 '79': 3,
                 'pound': 517,
                 'highest': 38,
                 '355': 1,
                 'super': 168,
                 'proud': 82,
                 'omg': 4,
                 'month': 967,
                 'fix': 88,
                 'around': 657,
                 '1500': 79,
                 'calori': 1440,
                 'use': 1030,
                 'scale': 260,
                 'mfp': 20,
                 'track': 321,
                 'mile': 214,
                 'walk': 449,
                 'train': 512,
                 '3': 795,
                 '45': 279,
                 'min': 541,
                 'ellipt': 27,
                 'trail': 9,
                 'saturday': 64,
                 'zumba': 4,
                 'sunday': 59,
                 'ive': 2078,
                 'gotten': 83,
                 'desper': 28,
                 'ask': 320,
                 'adipex': 3,
                 'see': 590,
                 'would': 1316,
                 'doesnt': 246,
                 'im': 4447,
                 'sure': 384,
                 'next': 272,
                 'got': 528,
                 'ta': 14,
                 'pleas': 294,
                 'answer': 141,
                 'men': 47,
                 'bodybuild': 53,
                 'waist': 54,
                 'trainer': 72,
                 'worri': 136,
                 'muscl': 696,
                 'loss': 500,
                 'hey': 176,
                 'basic': 200,
                 'old': 250,
                 'male': 94,
                 '6ft1': 2,
                 'weigh': 358,
                 '88kg': 4,
                 '10': 422,
                 'ago': 383,
                 'gym': 789,
                 '4': 444,
                 '90130': 1,
                 'g': 17,
                 'protein': 612,
                 'depend': 58,
                 'manag': 122,
                 'come': 331,
                 '78kg': 4,
                 'bodi': 1001,
                 'percentag': 40,
                 '1918': 1,
                 'percent': 10,
                 'accord': 55,
                 'boditrax': 1,
                 'machin': 123,
                 'accur': 63,
                 'follow': 217,
                 'trend': 17,
                 'show': 155,
                 'lost': 606,
                 '4kg': 5,
                 'sinc': 629,
                 'start': 1587,
                 'free': 124,
                 'mass': 97,
                 'drop': 143,
                 '3kg': 4,
                 'much': 1021,
                 'normal': 266,
                 'workout': 865,
                 'increas': 223,
                 'massiv': 26,
                 'deadlift': 104,
                 '40kg': 5,
                 '80kg': 11,
                 'calv': 36,
                 'rais': 145,
                 '85': 24,
                 'kg': 209,
                 'exampl': 104,
                 'shoukd': 1,
                 'trust': 38,
                 'seem': 383,
                 'consist': 234,
                 'hasnt': 41,
                 'gone': 94,
                 'appreci': 256,
                 'overthink': 11,
                 'let': 256,
                 'thank': 564,
                 'mount': 4,
                 'pull': 177,
                 'bar': 143,
                 'miss': 110,
                 'screw': 16,
                 'recent': 309,
                 'bought': 72,
                 'friend': 284,
                 'lead': 53,
                 'avoid': 116,
                 'death': 25,
                 'mean': 208,
                 'there': 157,
                 'fine': 115,
                 'barhttpswwwamazoncoukonetwofitmountedstableoutdoormaximumdpb07v66qq37_encodingutf8amppd_rd_whzrvdampcontentidamzn1sym31c9090
        c9b654f91bf3704dd92281172amppf_rd_p31c9090c9b654f91bf3704dd92281172amppf_rd_rq84s95zrx8pnwjzapzyhamppd_rd_wgasssvamppd_rd_r4c71ac1c2c1
        d486d8878a4aad9e61dcdampref_pd_gw_ci_mcx_mihttpswwwamazoncoukonetwofitmountedstableoutdoormaximumdpb07v66qq37_encodingutf8amppd_rd_whz
        rvdampcontentidamzn1sym31c9090c9b654f91bf3704dd92281172amppf_rd_p31c9090c9b654f91bf3704dd92281172amppf_rd_rq84s95zrx8pnwjzapzyhamppd_r
        d_wgasssvamppd_rd_r4c71ac1c2c1d486d8878a4aad9e61dcdampref_pd_gw_ci_mcx_mi': 1,
                 'alreadi': 208,
                 'differ': 361,
                 'mention': 73,
                 'ed': 21,
                 'sv': 13,
                 '18intersex': 1,
                 'first': 585,
                 '118': 3,
                 '260': 11,
                 'lb': 519,
                 'sad': 45,
                 'anyth': 368,
                 'motiv': 203,
                 'journey': 230,
                 '113': 1,
                 '249': 2,
                 'remark': 1,
                 'easier': 78,
                 'past': 327,
                 'attempt': 61,
                 'pleasantli': 3,
                 'surpris': 41,
                 'get': 2008,
                 'hate': 152,
                 'could': 527,
                 'form': 151,
                 'coher': 1,
                 'thought': 375,
                 'turn': 135,
                 'healthiest': 15,
                 'way': 666,
                 'easiest': 7,
                 'albeit': 3,
                 'slowest': 4,
                 'starv': 46,
                 'think': 901,
                 'look': 1010,
                 'envi': 3,
                 'still': 759,
                 'struggl': 224,
                 'one': 954,
                 'tbh': 18,
                 'far': 188,
                 'face': 107,
                 'defin': 19,
                 'okay': 85,
                 'definit': 148,
                 'genet': 52,
                 'bone': 63,
                 'structur': 32,
                 'thing': 647,
                 'cheekbon': 2,
                 'pronounc': 5,
                 'also': 990,
                 'favorit': 80,
                 'pair': 47,
                 'jean': 38,
                 'looser': 6,
                 'thigh': 63,
                 'great': 252,
                 'fit': 506,
                 'kind': 201,
                 'word': 89,
                 'post': 406,
                 'posit': 82,
                 'benefici': 10,
                 'wish': 56,
                 'everyon': 285,
                 'luck': 19,
                 'cocacola': 2,
                 'make': 1031,
                 'skinni': 146,
                 'junk': 79,
                 'liter': 89,
                 'cola': 2,
                 '185': 19,
                 'cm': 32,
                 '60': 132,
                 'quit': 243,
                 'went': 347,
                 '70': 38,
                 'want': 1591,
                 'sick': 84,
                 'cant': 579,
                 'h': 12,
                 'cope': 15,
                 'happen': 210,
                 'deficit': 277,
                 'exercis': 875,
                 'regularli': 50,
                 'stupid': 34,
                 'homework': 3,
                 'lol': 118,
                 'higher': 70,
                 'intak': 160,
                 'lower': 227,
                 'least': 208,
                 'dont': 1690,
                 'immedi': 51,
                 'back': 1080,
                 'minut': 588,
                 '1000': 39,
                 'shit': 68,
                 'metabol': 89,
                 'what': 121,
                 'two': 363,
                 'mum': 13,
                 'put': 296,
                 'oil': 115,
                 'soemth': 1,
                 'lmfao': 1,
                 'anyon': 712,
                 'els': 275,
                 'experienc': 57,
                 'rebuild': 4,
                 'routin': 401,
                 'lss': 1,
                 'long': 452,
                 'stori': 92,
                 'short': 132,
                 'pretti': 379,
                 'crossfit': 96,
                 'covid': 78,
                 '5x': 17,
                 '2x': 20,
                 'sprain': 5,
                 'ankl': 41,
                 'pt': 32,
                 'releas': 21,
                 'juli': 39,
                 'find': 462,
                 'demotiv': 12,
                 'stress': 127,
                 'balanc': 81,
                 'strategi': 52,
                 'found': 173,
                 'build': 256,
                 'ever': 241,
                 'made': 330,
                 'video': 100,
                 'cbum': 1,
                 'say': 487,
                 'btch': 1,
                 'connect': 17,
                 'hrm': 4,
                 'non': 42,
                 'tablet': 13,
                 'rower': 68,
                 'occasion': 76,
                 'visit': 32,
                 'studio': 60,
                 'home': 322,
                 'otf': 183,
                 '1kg': 9,
                 '2lb': 9,
                 'hi': 315,
                 '13yearold': 1,
                 'unabl': 20,
                 'felt': 232,
                 'imposs': 37,
                 'grandma': 10,
                 'almost': 226,
                 'crazi': 76,
                 'portion': 103,
                 'person': 258,
                 'smaller': 57,
                 'scold': 1,
                 'enjoy': 163,
                 'take': 642,
                 'tell': 249,
                 'ye': 55,
                 'asian': 13,
                 'insan': 31,
                 'per': 335,
                 '58': 31,
                 'larg': 97,
                 'bowl': 51,
                 'hard': 388,
                 'convinc': 33,
                 'parent': 64,
                 'buy': 205,
                 'ton': 67,
                 'brother': 32,
                 'told': 152,
                 'number': 149,
                 'benefit': 56,
                 'famili': 162,
                 'overweight': 134,
                 'physic': 148,
                 'everyday': 127,
                 'enough': 344,
                 'fight': 37,
                 'urg': 16,
                 'boredom': 12,
                 'overeat': 16,
                 'tast': 90,
                 'someth': 535,
                 'ill': 332,
                 'control': 136,
                 'feel': 1684,
                 'unsatisfi': 3,
                 'frustrat': 112,
                 'lack': 67,
                 'progress': 388,
                 'shape': 103,
                 'advanc': 111,
                 'tip': 234,
                 'jog': 30,
                 'speed': 50,
                 'orangetheori': 20,
                 'runner': 32,
                 'previous': 33,
                 'steadili': 18,
                 '1012': 9,
                 'base': 522,
                 'pace': 85,
                 'inclin': 111,
                 'push': 609,
                 'ao': 239,
                 'templat': 81,
                 'maintain': 162,
                 'overal': 81,
                 'especi': 171,
                 'improv': 155,
                 'everi': 462,
                 'class': 341,
                 'arent': 89,
                 'quad': 32,
                 'leg': 401,
                 '52': 37,
                 'mediumshort': 1,
                 'height': 90,
                 'wonder': 288,
                 'attract': 22,
                 'proport': 11,
                 'thoughtsexperi': 2,
                 'chocol': 62,
                 'mani': 348,
                 'slightli': 66,
                 'silli': 15,
                 'question': 457,
                 'approxim': 19,
                 '550': 2,
                 '100': 130,
                 'gram': 69,
                 'matter': 131,
                 'dark': 17,
                 'milk': 167,
                 'white': 59,
                 'read': 198,
                 'ingredi': 89,
                 'cocoa': 6,
                 '228': 3,
                 'cal': 126,
                 'sugar': 189,
                 '387': 1,
                 '42': 6,
                 '884': 1,
                 'idea': 287,
                 'everyth': 243,
                 'except': 80,
                 'process': 96,
                 'product': 176,
                 'individu': 18,
                 'imagin': 40,
                 'option': 197,
                 'five': 35,
                 'yet': 147,
                 '53': 31,
                 '21': 53,
                 'f': 43,
                 '180': 35,
                 'count': 261,
                 'loseit': 18,
                 'app': 170,
                 'stay': 294,
                 'treadmil': 118,
                 '46': 17,
                 'wan': 91,
                 'na': 158,
                 '165': 20,
                 'anniversari': 3,
                 'mid': 45,
                 'novemb': 19,
                 'gon': 66,
                 'happi': 184,
                 'done': 228,
                 '83': 3,
                 'updat': 47,
                 'alway': 477,
                 'obes': 80,
                 'anorex': 5,
                 'age': 80,
                 '17': 38,
                 '110lb': 7,
                 '59': 43,
                 'bdd': 1,
                 'slim': 44,
                 '18': 65,
                 '20': 225,
                 '135145lb': 1,
                 'stop': 368,
                 'restrict': 75,
                 'ate': 150,
                 'fast': 299,
                 'bit': 427,
                 'hit': 225,
                 'symptom': 14,
                 'certain': 65,
                 'issu': 272,
                 'battl': 12,
                 'psychosi': 1,
                 'physiolog': 2,
                 '102lb': 1,
                 'heaviest': 29,
                 '237lb': 3,
                 'antipsychot': 1,
                 'didnt': 352,
                 'either': 172,
                 'anyway': 135,
                 'june': 44,
                 'particular': 26,
                 'incid': 6,
                 'click': 46,
                 'realiz': 108,
                 'never': 459,
                 'spectrum': 3,
                 'opt': 6,
                 'approach': 55,
                 'join': 121,
                 'noom': 13,
                 'sponsor': 2,
                 'log': 67,
                 'emot': 50,
                 '25lb': 5,
                 'middl': 25,
                 'begin': 100,
                 'septemb': 89,
                 'camp': 7,
                 'trip': 31,
                 'anti': 8,
                 'depress': 121,
                 '10lb': 24,
                 'anymor': 123,
                 '118lb': 3,
                 'plan': 316,
                 'point': 284,
                 'dress': 27,
                 'mom': 67,
                 'wed': 36,
                 'summer': 55,
                 'tldr': 36,
                 'anorexia': 12,
                 'coupl': 203,
                 '101lb': 1,
                 '19lb': 2,
                 '135lb': 7,
                 'squat': 381,
                 'lighthead': 5,
                 'stick': 151,
                 'press': 201,
                 'experi': 192,
                 'close': 129,
                 'bedtim': 4,
                 '254': 2,
                 'caus': 195,
                 'addit': 59,
                 'myth': 11,
                 '700': 22,
                 '17502000': 1,
                 'late': 123,
                 'shower': 16,
                 'mainten': 120,
                 '300': 24,
                 'rich': 19,
                 'bag': 77,
                 'dri': 67,
                 'pineappl': 4,
                 'bed': 72,
                 'replac': 63,
                 'oat': 52,
                 'chicken': 247,
                 'peanut': 31,
                 'safest': 2,
                 'absolut': 81,
                 'total': 204,
                 'eliminationa': 1,
                 'carbveggiefruit': 1,
                 'sourc': 64,
                 'choos': 64,
                 'categori': 17,
                 'refreez': 4,
                 'begond': 1,
                 'brick': 6,
                 'stock': 17,
                 'beyond': 32,
                 'beef': 56,
                 'refridger': 1,
                 'section': 35,
                 'sale': 17,
                 'bec': 2,
                 'bacteria': 3,
                 'degrad': 5,
                 'true': 61,
                 'safe': 56,
                 'sauc': 72,
                 'amp': 265,
                 'chilli': 5,
                 'patti': 3,
                 'textur': 23,
                 'might': 193,
                 'ok': 84,
                 'bunch': 49,
                 'freez': 33,
                 'strength': 323,
                 'without': 302,
                 'mess': 50,
                 'id': 370,
                 'pullup': 86,
                 'weaken': 2,
                 'arm': 235,
                 'up': 208,
                 'may': 218,
                 'averag': 136,
                 'well': 458,
                 'compar': 78,
                 'refer': 45,
                 'assist': 27,
                 'focu': 145,
                 'em': 8,
                 'weaker': 17,
                 'aspect': 21,
                 'full': 215,
                 'skill': 41,
                 'practic': 65,
                 'access': 48,
                 'hero': 8,
                 'wod': 45,
                 'typic': 58,
                 'tough': 61,
                 'modifi': 16,
                 'shorter': 24,
                 'side': 190,
                 'tg': 1,
                 'candid': 8,
                 'ear': 13,
                 'recommend': 260,
                 'vein': 14,
                 'fell': 32,
                 'left': 148,
                 'exterior': 1,
                 'calf': 44,
                 'bottom': 25,
                 '38': 13,
                 'titl': 67,
                 'realis': 26,
                 '190lb': 10,
                 'big': 263,
                 'set': 404,
                 'mileston': 30,
                 'along': 63,
                 'today': 563,
                 'reddit': 48,
                 'alon': 52,
                 'share': 342,
                 'success': 83,
                 'interest': 127,
                 'cut': 292,
                 'empti': 33,
                 'live': 244,
                 'kept': 55,
                 'aim': 54,
                 '1000day': 1,
                 '220lb': 5,
                 'finger': 31,
                 'cross': 31,
                 'bing': 158,
                 'tini': 27,
                 'vacat': 39,
                 '3k': 4,
                 'strenght': 10,
                 'ham': 15,
                 'macro': 105,
                 '23': 90,
                 'kilo': 27,
                 'subtl': 1,
                 'measur': 95,
                 'hopeless': 20,
                 'miser': 27,
                 'ruin': 39,
                 'continu': 161,
                 'recomp': 34,
                 'mayb': 315,
                 'intuit': 14,
                 'religi': 17,
                 'pal': 11,
                 'platform': 6,
                 'samsung': 8,
                 'goe': 70,
                 'target': 46,
                 'near': 53,
                 '1850150': 1,
                 '1633': 2,
                 'add': 219,
                 'whatev': 113,
                 'fitnessexercis': 2,
                 'earn': 7,
                 'lift': 511,
                 '40': 101,
                 '600': 20,
                 '2233': 1,
                 'wherea': 8,
                 '1850': 2,
                 'm21': 1,
                 '163': 3,
                 'impost': 1,
                 'syndrom': 5,
                 'mirror': 43,
                 'isnt': 215,
                 'weird': 85,
                 'thursday': 57,
                 'submit': 3,
                 'check': 165,
                 'program': 239,
                 'saw': 71,
                 'growth': 34,
                 'anadrom': 1,
                 '50': 102,
                 'medic': 93,
                 'buydu': 1,
                 'legit': 8,
                 'new': 422,
                 'world': 73,
                 'curiou': 87,
                 'meet': 38,
                 'man': 58,
                 '6': 439,
                 'ft': 13,
                 '195': 9,
                 '10000': 17,
                 'step': 197,
                 'doubl': 28,
                 'prework': 2,
                 'women': 77,
                 'taken': 57,
                 'tdee': 99,
                 'bmrrmr': 1,
                 'onlin': 99,
                 'calcul': 105,
                 'regard': 45,
                 'burn': 172,
                 'resourc': 37,
                 'best': 285,
                 'estim': 29,
                 'throughout': 78,
                 'harder': 68,
                 'actual': 331,
                 'complet': 212,
                 'odd': 20,
                 'round': 174,
                 'sedentari': 61,
                 'bmr': 22,
                 'cool': 48,
                 'lay': 31,
                 'coma': 1,
                 'clean': 110,
                 'movement': 92,
                 'etc': 308,
                 'neat': 5,
                 'thermic': 1,
                 '200300': 4,
                 'wat': 1,
                 'account': 276,
                 'top': 111,
                 '15000': 3,
                 '17000': 1,
                 'reliabl': 8,
                 'seemingli': 6,
                 'credibl': 1,
                 'watch': 136,
                 'strap': 15,
                 'wrist': 42,
                 'abund': 2,
                 'octob': 72,
                 'benchmark': 26,
                 'throw': 59,
                 'anxieti': 55,
                 'stomach': 138,
                 'anxiou': 19,
                 'limit': 126,
                 'rariti': 1,
                 'regain': 15,
                 'scare': 72,
                 'high': 349,
                 'daili': 459,
                 'threw': 15,
                 'pizza': 51,
                 'whole': 209,
                 'asid': 27,
                 'guilti': 28,
                 '30min': 10,
                 'cocain': 1,
                 'overwhelm': 25,
                 'panic': 11,
                 'eaten': 49,
                 'suddenli': 27,
                 'rid': 36,
                 'toilet': 6,
                 'coke': 8,
                 'guess': 146,
                 'roast': 43,
                 'carb': 202,
                 'panik': 1,
                 'randomli': 18,
                 'bathroom': 10,
                 'wouldnt': 83,
                 'tomorrow': 68,
                 'bad': 342,
                 'garlic': 39,
                 'wrap': 17,
                 'sensat': 6,
                 'fatter': 10,
                 'alley': 1,
                 'morbidli': 8,
                 'pressur': 32,
                 'poor': 32,
                 'excus': 27,
                 'unemploy': 4,
                 'gf': 10,
                 'anywher': 29,
                 'hell': 87,
                 'wrong': 183,
                 'anxietydrinkingdrug': 1,
                 'usetoo': 1,
                 'purpos': 25,
                 'bulimia': 5,
                 'ref': 2,
                 '26': 46,
                 '62': 24,
                 '194lb': 4,
                 '330': 7,
                 'unpleas': 1,
                 'smell': 29,
                 'attend': 29,
                 'member': 93,
                 'becam': 40,
                 'frequent': 41,
                 'particip': 20,
                 'strong': 83,
                 'odor': 1,
                 'wors': 85,
                 'front': 101,
                 'desk': 23,
                 'remind': 27,
                 'facebook': 5,
                 'instagram': 8,
                 'sign': 49,
                 'shrink': 15,
                 'seen': 120,
                 'board': 12,
                 'oh': 41,
                 'neg': 45,
                 'tornado': 22,
                 'becom': 155,
                 'deal': 94,
                 'unwelcom': 1,
                 'deserv': 11,
                 'shake': 73,
                 'appl': 61,
                 'meal': 590,
                 'troubl': 59,
                 'planningrememb': 1,
                 'eatfeel': 1,
                 'regular': 66,
                 'forc': 65,
                 'chug': 2,
                 'caffein': 24,
                 'honest': 33,
                 'questionpleas': 1,
                 'asshol': 3,
                 'vegan': 889,
                 'lifestyl': 100,
                 'relat': 67,
                 'truli': 34,
                 'impact': 50,
                 'wont': 160,
                 'anyjust': 1,
                 'genuin': 44,
                 'disord': 68,
                 '16ftm': 2,
                 '169cm': 2,
                 'sw': 53,
                 '68': 23,
                 'cw': 56,
                 '49': 4,
                 'cheat': 75,
                 'half': 204,
                 '750': 2,
                 'monday': 70,
                 '400': 16,
                 'uncomfort': 43,
                 'reason': 212,
                 'unreli': 2,
                 'nondigit': 1,
                 '15': 262,
                 'horror': 3,
                 'weighin': 69,
                 'school': 142,
                 'nurs': 6,
                 '499': 1,
                 'pant': 40,
                 'cri': 33,
                 'expect': 76,
                 'tmi': 4,
                 'poop': 7,
                 'serious': 66,
                 'hope': 266,
                 'underweight': 33,
                 'mostli': 151,
                 'allergi': 10,
                 'condit': 42,
                 'veggi': 121,
                 'tire': 149,
                 'carrot': 26,
                 'celeri': 7,
                 'cucumb': 17,
                 'snap': 15,
                 'pea': 22,
                 'crunch': 26,
                 'kinda': 78,
                 'brain': 45,
                 'chew': 19,
                 'tasti': 20,
                 ...})
```

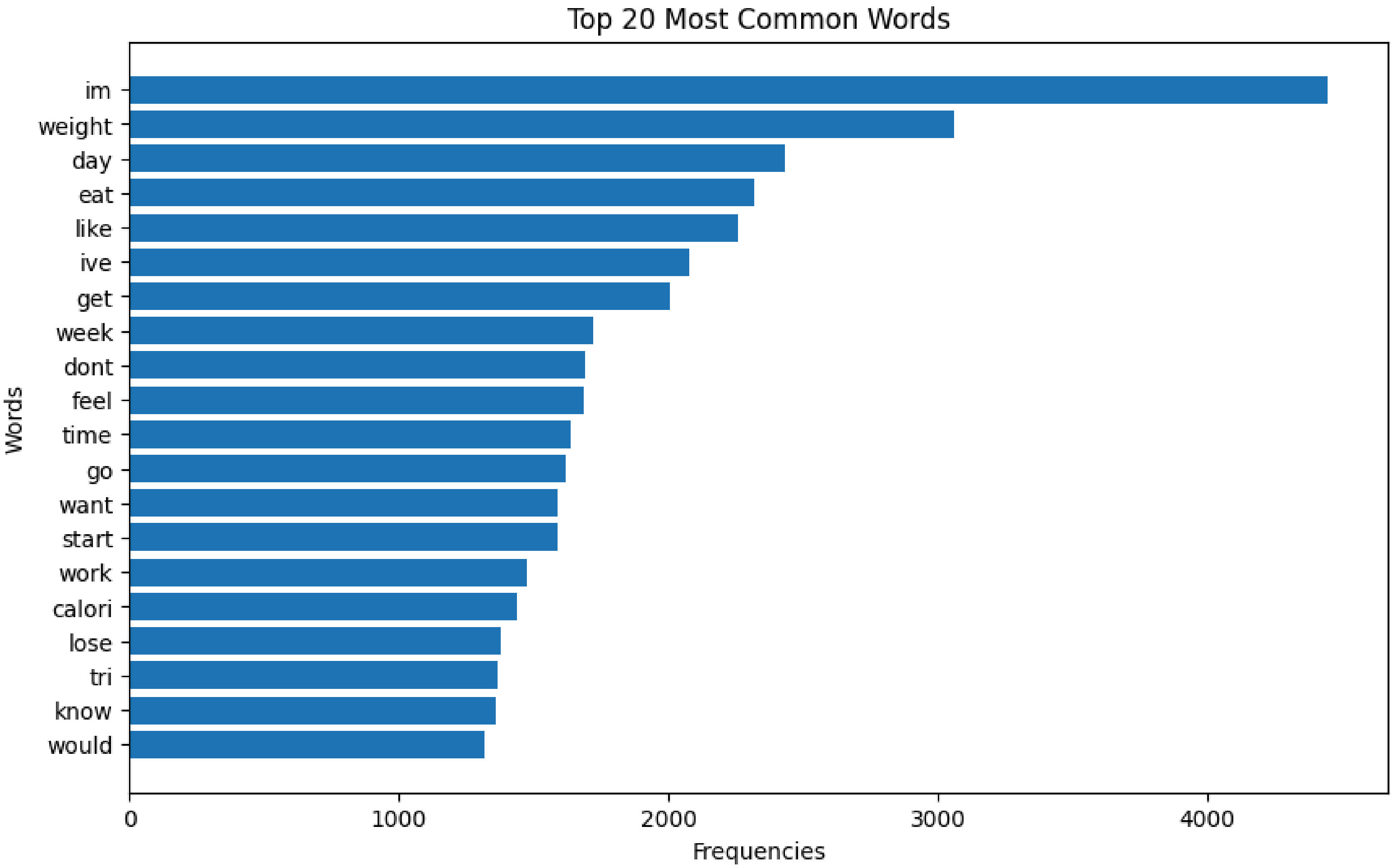
In [ ]:

```
# Top 20 words
most_common_words = word_counts.most_common(20)
words, frequencies =zip(*most_common_words)
print(words)
print(frequencies)
```

```
('im', 'weight', 'day', 'eat', 'like', 'ive', 'get', 'week', 'dont', 'feel', 'time', 'go', 'want', 'start', 'work', 'calori', 'lose',
'tri', 'know', 'would')
(4447, 3058, 2436, 2316, 2257, 2078, 2008, 1720, 1690, 1684, 1636, 1620, 1591, 1587, 1475, 1440, 1380, 1369, 1360, 1316)
```

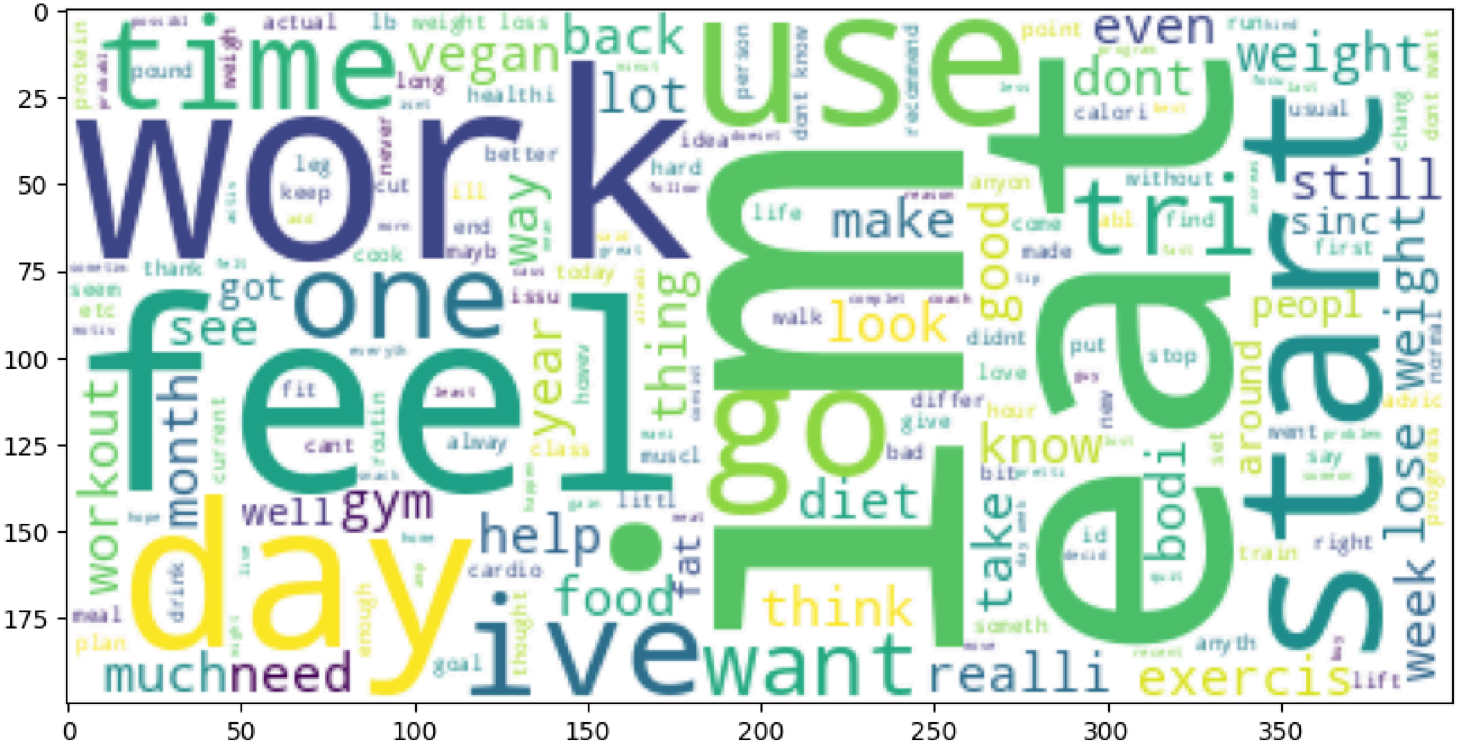
In [ ]:

```
# Word frequency visualization
# Word frequency visualization
plt.figure(figsize = (10, 6))
plt.barh(words, frequencies)
plt.xlabel('Frequencies')
plt.ylabel('Words')
plt.title('Top 20 Most Common Words')
plt.gca() .invert_yaxis()
plt. show()
```



In [ ]:

```
# Word Cloud
all_words = [word for list in df['processed'] for word in list]
text = ' '.join(all_words)
wordcloud = WordCloud(max_words = 200, background_color = 'white').generate(text)
plt.figure(figsize = (10,5))
plt.imshow(wordcloud)
plt.show()
```



```
In [ ]: text
```

```
In [ ]: #create a dictionary
        dictionary = corpora.Dictionary(df['processed'])
        #create a corpus
        corpus = [dictionary.doc2bow(text) for text in df['processed']]
```

```
In [ ]: # train LDA, Topic = 2
        lda_model_2 = LdaModel(corpus = corpus, id2word = dictionary, num_topics = 2, random_state = 42)
        # topic word matrix
        for idx, topic in lda_model_2.print_topics (-1, num_words = 10):
          print('Topic: {} \nwords: {}'.format(idx, topic))
```

```
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the number of passes or iterations to
improve accuracy
Topic: 0
words: 0.017*"im" + 0.013*"weight" + 0.012*"day" + 0.009*"eat" + 0.008*"get" + 0.008*"like" + 0.007*"week" + 0.007*"feel" + 0.007*"sta
rt" + 0.007*"calori"
Topic: 1
words: 0.014*"im" + 0.010*"ive" + 0.008*"weight" + 0.008*"like" + 0.008*"eat" + 0.006*"time" + 0.006*"work" + 0.006*"go" + 0.006*"don
t" + 0.006*"get"
```

```
In [ ]: # train LDA, Topic = 4
        lda_model_4 = LdaModel(corpus = corpus, id2word = dictionary, num_topics = 4, random_state = 42)
        # topic word matrix
        for idx, topic in lda_model_4.print_topics (-1, num_words = 10):
          print('Topic: {} \nwords: {}'.format(idx, topic))
```

```
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the number of passes or iterations to
improve accuracy
Topic: 0
words: 0.018*"im" + 0.013*"weight" + 0.013*"day" + 0.009*"get" + 0.008*"like" + 0.007*"start" + 0.007*"feel" + 0.006*"dont" + 0.006*"f
at" + 0.006*"eat"
Topic: 1
words: 0.014*"im" + 0.008*"like" + 0.008*"weight" + 0.006*"ive" + 0.006*"go" + 0.006*"get" + 0.006*"time" + 0.006*"dont" + 0.006*"bod
i" + 0.006*"tri"
Topic: 2
words: 0.011*"im" + 0.008*"1" + 0.007*"like" + 0.007*"min" + 0.007*"sec" + 0.007*"time" + 0.007*"2" + 0.006*"row" + 0.006*"3" + 0.006
*"week"
Topic: 3
words: 0.019*"im" + 0.015*"weight" + 0.014*"eat" + 0.012*"ive" + 0.010*"day" + 0.009*"want" + 0.008*"like" + 0.008*"lose" + 0.008*"wee
k" + 0.008*"calori"
```

```
In [ ]: # train LDA, Topic = 8
        lda_model_8 = LdaModel(corpus = corpus, id2word = dictionary, num_topics = 8, random_state = 42)
        # topic word matrix
        for idx, topic in lda_model_8.print_topics (-1, num_words = 10):
          print('Topic: {} \nwords: {}'.format(idx, topic))
```

```
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the number of passes or iterations to
improve accuracy
```

```
Topic: 0
words: 0.012*"im" + 0.010*"day" + 0.008*"question" + 0.007*"like" + 0.007*"get" + 0.007*"account" + 0.006*"share" + 0.006*"daili" + 0.
005*"dont" + 0.005*"thread"
Topic: 1
words: 0.011*"im" + 0.010*"vegan" + 0.008*"time" + 0.007*"like" + 0.006*"anyon" + 0.006*"go" + 0.005*"tri" + 0.004*"get" + 0.004*"wor
k" + 0.004*"need"
Topic: 2
words: 0.015*"sec" + 0.015*"1" + 0.014*"min" + 0.013*"block" + 0.013*"x" + 0.013*"row" + 0.011*"base" + 0.011*"push" + 0.010*"30" + 0.
010*"minut"
Topic: 3
words: 0.015*"vegan" + 0.014*"im" + 0.010*"eat" + 0.009*"work" + 0.009*"want" + 0.008*"like" + 0.007*"ive" + 0.007*"food" + 0.006*"da
y" + 0.006*"anim"
Topic: 4
words: 0.018*"im" + 0.016*"eat" + 0.014*"like" + 0.011*"dont" + 0.010*"feel" + 0.009*"calori" + 0.009*"protein" + 0.008*"get" + 0.007
*"day" + 0.007*"fat"
Topic: 5
words: 0.019*"weight" + 0.014*"im" + 0.012*"week" + 0.012*"ive" + 0.010*"eat" + 0.010*"start" + 0.009*"day" + 0.009*"lose" + 0.009*"ge
t" + 0.009*"like"
Topic: 6
words: 0.022*"im" + 0.013*"weight" + 0.010*"eat" + 0.010*"day" + 0.007*"calori" + 0.007*"ive" + 0.007*"diet" + 0.007*"feel" + 0.006*"t
ri" + 0.006*"get"
Topic: 7
words: 0.022*"im" + 0.015*"weight" + 0.012*"day" + 0.009*"dont" + 0.008*"ive" + 0.008*"get" + 0.008*"like" + 0.008*"feel" + 0.007*"sta
rt" + 0.007*"eat"
```

In [ ]:

```python
# LDA evaluation
# Coherence
coherence_mode11_1da = CoherenceModel (model = lda_model_2, texts = df['processed'],
                                       dictionary = dictionary, coherence = 'c_v').get_coherence()

coherence_mode12_1da = CoherenceModel (model = lda_model_4, texts = df['processed'],
                                       dictionary = dictionary, coherence = 'c_v').get_coherence()

coherence_mode13_1da = CoherenceModel (model = lda_model_8, texts = df['processed'],
                                       dictionary = dictionary, coherence = 'c_v').get_coherence()


print('\nCoherence score 1', coherence_mode11_1da)
print('InCoherence score 2', coherence_mode12_1da)
print('\nCoherence score 3', coherence_mode13_1da)
```

```
Coherence score 1 0.5352799718462801
InCoherence score 2 0.4897607824133229

Coherence score 3 0.5352799718462801
```

Sentiment Analysis

In [ ]:

```python
# Define sentiment analysis function
analyzer = SentimentIntensityAnalyzer()
```

```
def get_post_sentiment(text):
  return analyzer. polarity_scores (text)['compound'] # return sentiment of a tweet
df['sentiment_score_vader'] = df['Titles_Body'].apply(get_post_sentiment)
df.head()
```

Out[ ]:

	Reddit_ID	Subreddits	Titles	Body	Author	Initial Create	Retrieved_date	Date_of_collection	Number of Comments	Decision	Titles_Body	pro
0	xn3aad	loseit	Why am I gaining weight on a healthy diet?	Hello, \nI really need some advice for someon...	Shinigami_Sadies	9/24/2022 16:33	9/24/2022 16:33	2022_9_23	29.0	0.0	Why am I gaining weight on a healthy diet? Hel...	die re
1	xyz6l9	xxfitness	I ran a sub 30 5K	I know that’s easy for most people but I haven...	Choice_Ad522	10/8/2022 13:59	10/8/2022 13:59	2022_10_7	3.0	1.0	I ran a sub 30 5K I know that’s easy for most ...	[ra th
2	x6u9cp	loseit	I am losing my mind instead of weight &gt;:(	I am currently 276, so 79 pounds down from my ...	peppermintwhitemocha	9/5/2022 19:09	9/5/2022 19:09	2022_9_4	22.0	0.0	I am losing my mind instead of weight &gt;:(I...	i wei (
3	ybse4o	bodybuilding	Please answer!!!	Do men bodybuilders use waist trainers?????	GrowthMobile	10/23/2022 16:41	10/23/2022 16:42	2022_10_22	12.0	1.0	Please answer!!! Do men bodybuilders use waist...	boc use
4	x0yt2o	loseit	Worried about muscle loss	Hey\nSo basically I’m a 30 year old male Im 6f...	Takedownkd	8/29/2022 16:22	8/29/2022 16:22	2022_8_28	4.0	0.0	Worried about muscle loss Hey\nSo basically I’...	lo ba 30

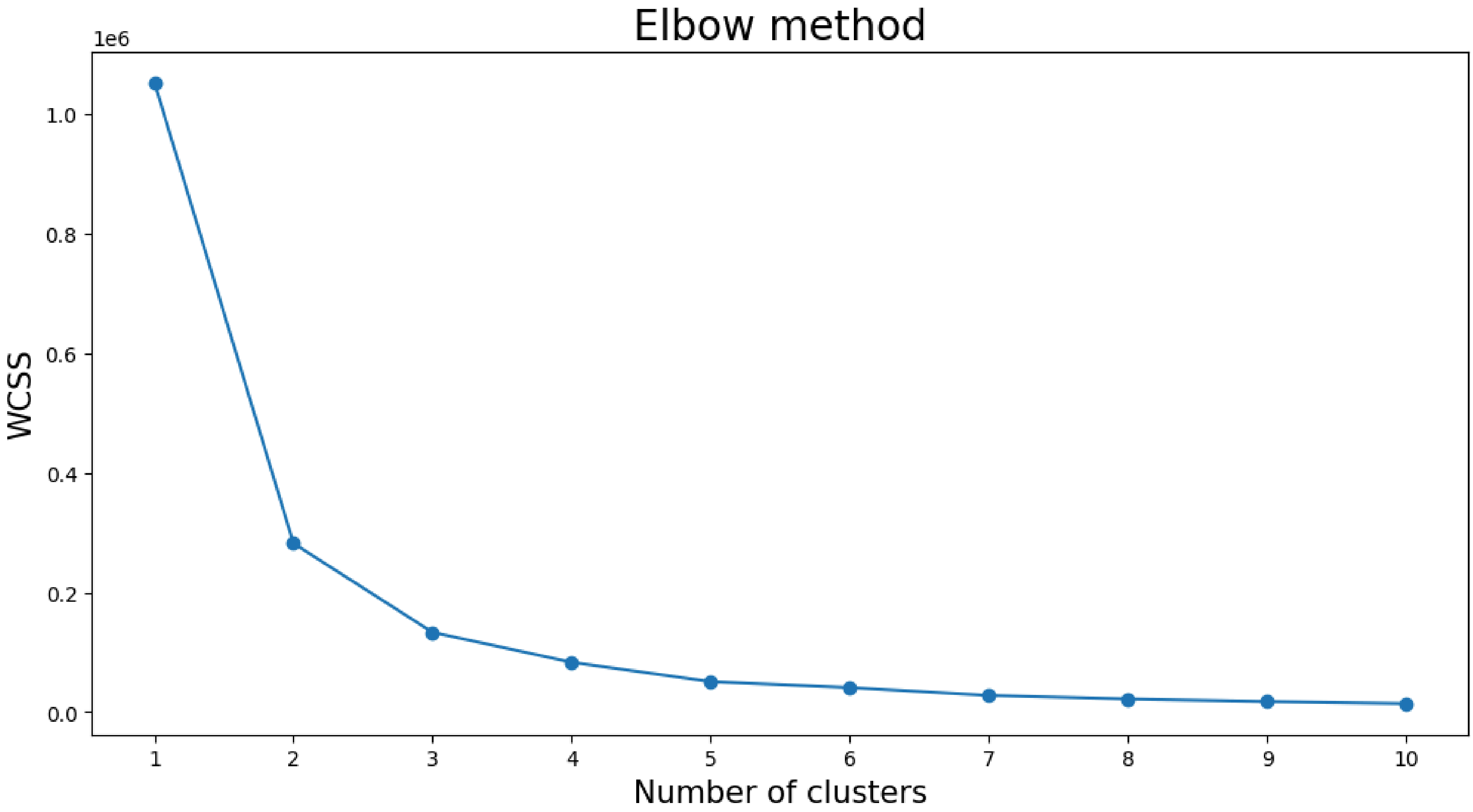
In [ ]:

```
#K-mean Clustering
```

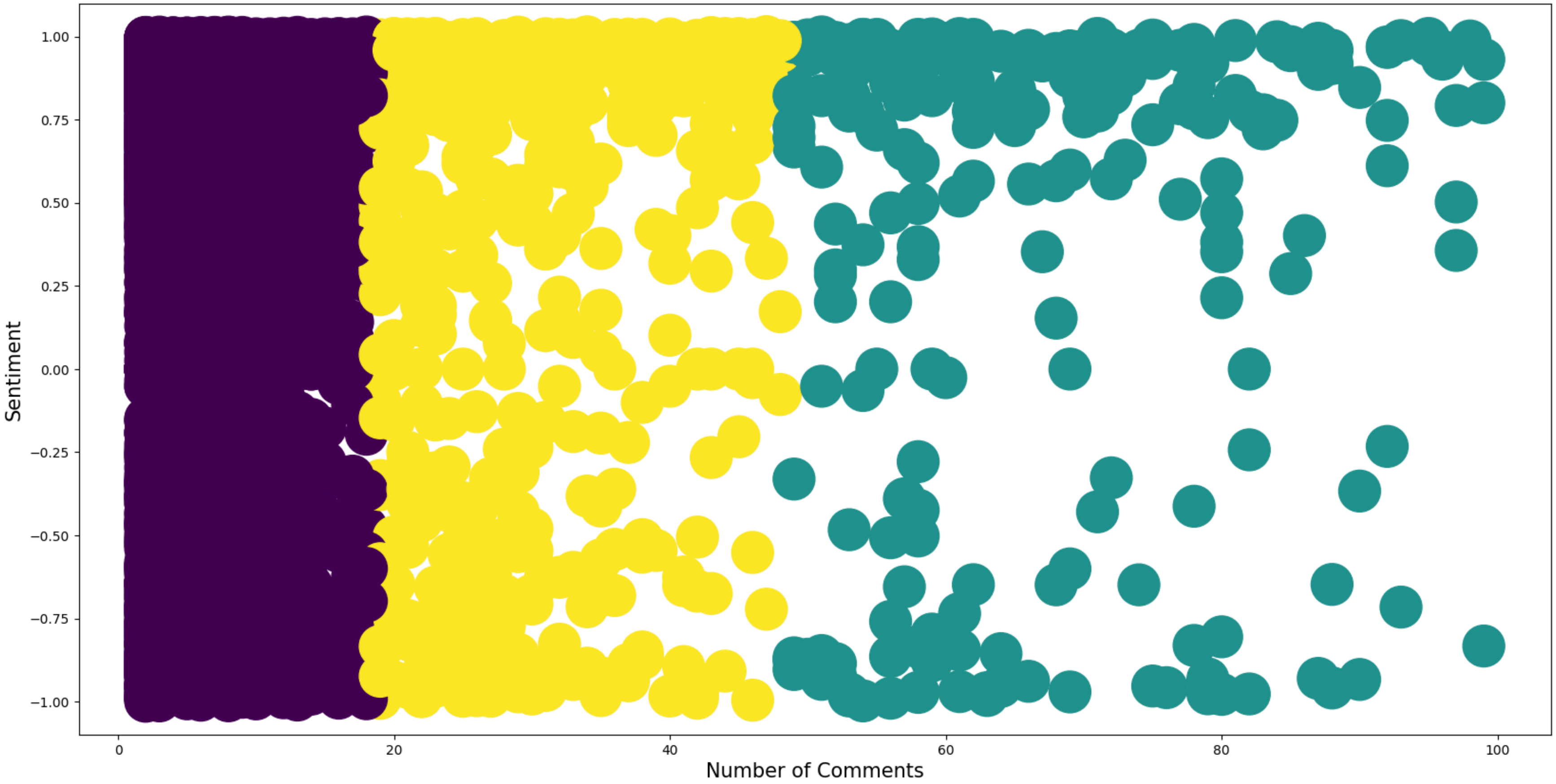
In [ ]:

```
data = list(zip(df['Number of Comments'], df['sentiment_score_vader']))
```

```
inertias = []
for i in range (1,11):
  kmeans = KMeans(n_clusters=i)
  kmeans.fit(data)
  inertias. append (kmeans. inertia_)


plt.figure(figsize=(12,6))
plt.plot(range (1,11), inertias, marker='o')
plt.xticks(np.arange(1, 11, 1.0),fontsize = 10)
plt.yticks(fontsize = 10)
plt.title('Elbow method',fontsize = 20)
plt.xlabel('Number of clusters',fontsize = 15)
plt.ylabel('WCSS',fontsize = 15)
plt.show()
```



```
In [ ]: kmeans = KMeans(n_clusters=3)
        kmeans.fit(data)
        plt.figure(figsize=(20,10))
        plt.scatter(df['Number of Comments'], df['sentiment_score_vader'], c=kmeans.labels_, s=1000)
        plt.xlabel('Number of Comments', fontsize = 15)
        plt.ylabel('Sentiment', fontsize = 15)
        plt.xticks(fontsize = 10)
        plt.yticks(fontsize = 10)
        plt.show()
```



Hierarchcal clustering

```
In [ ]: distance_matrix = pdist(df[['Number of Comments', 'sentiment_score_vader']], metric = "euclidean")
        distance_matrix
```

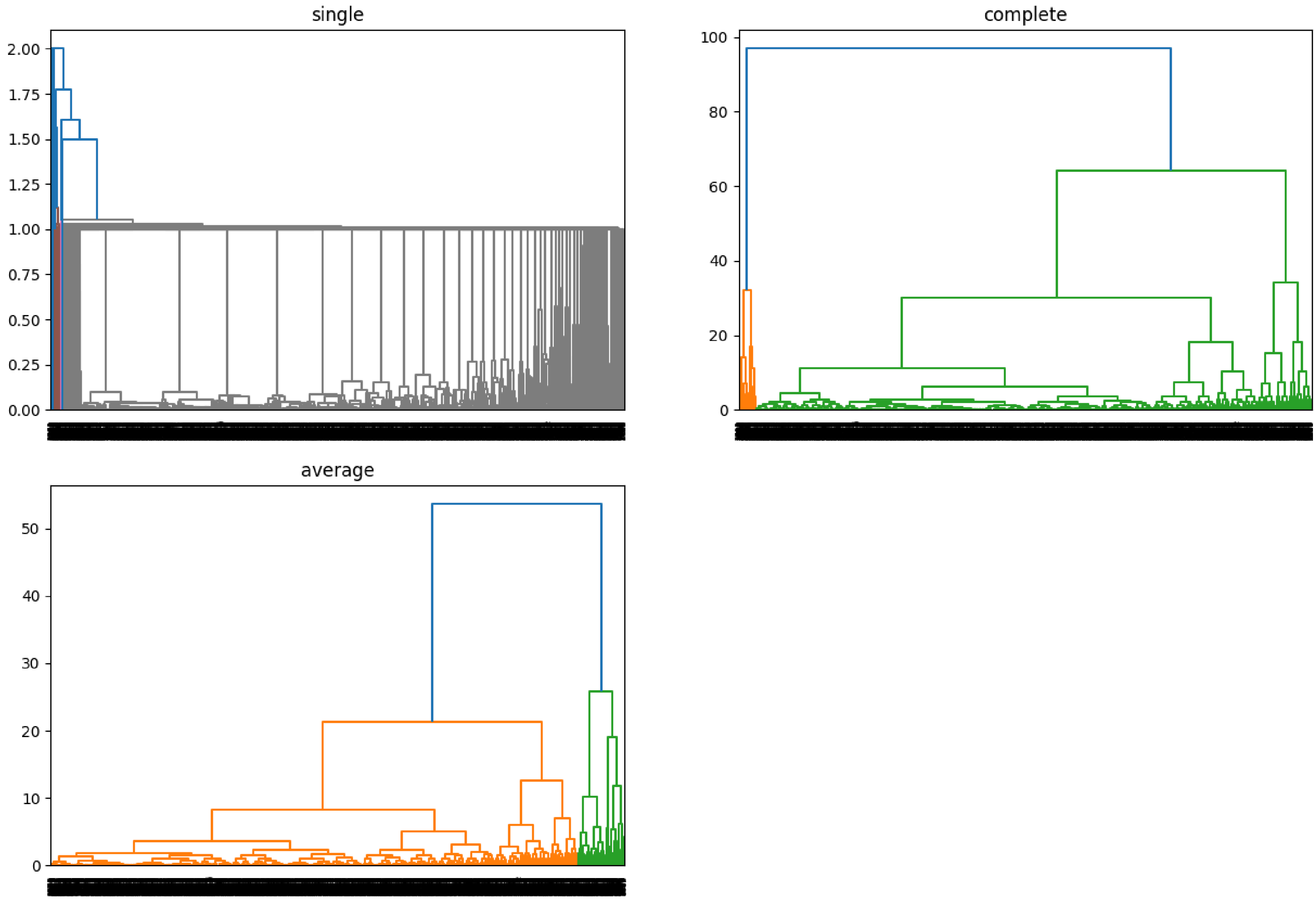
```
Out[ ]: array([26.01016203,  7.01659556, 17.00331817, ..., 31.01471275,
                1.17919464, 32.00170461])
```

In [ ]:

```
H_single = linkage(distance_matrix, method = 'single')
H_complete = linkage(distance_matrix, method = 'complete')
H_average = linkage(distance_matrix, method = 'average')
```

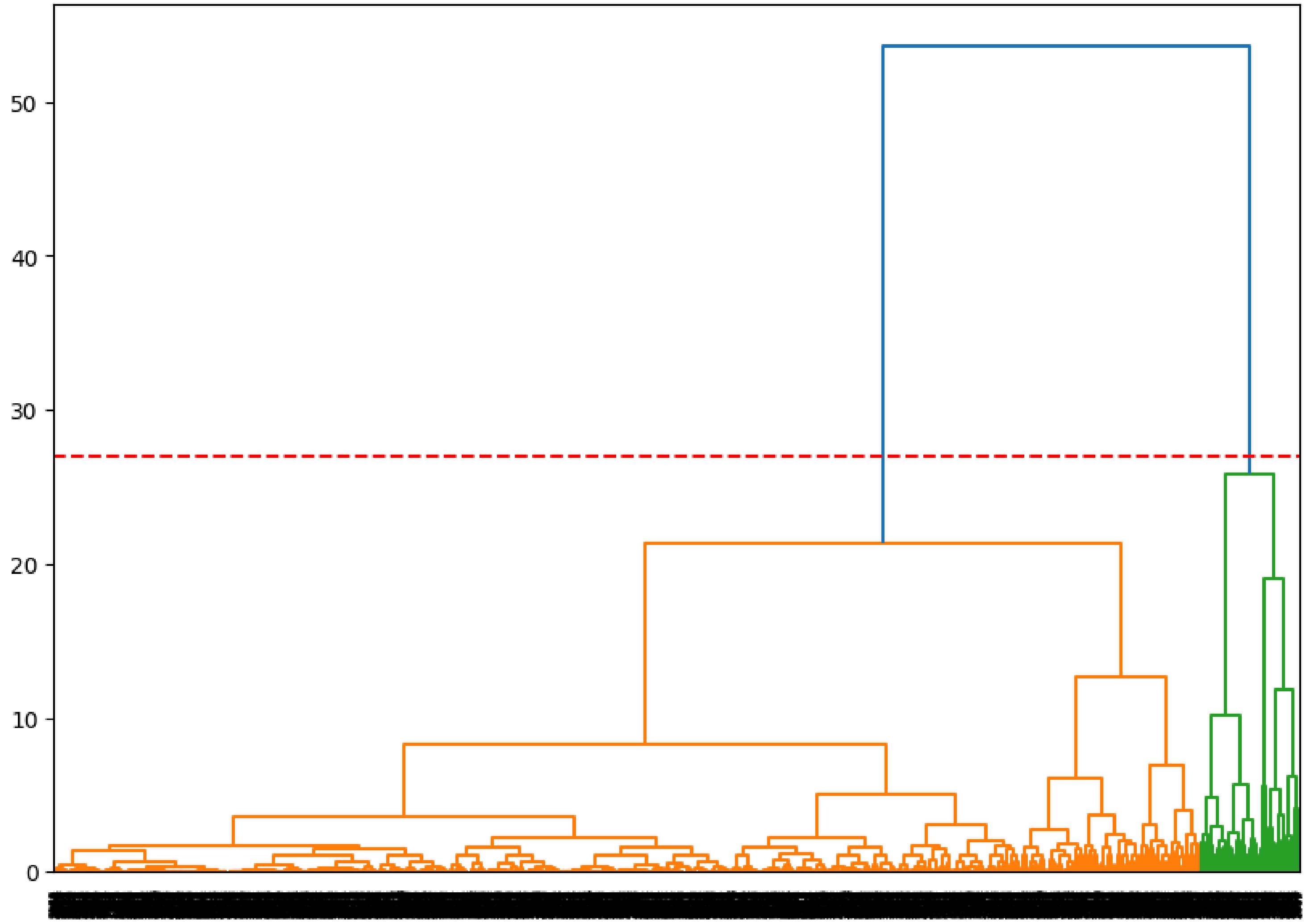
In [ ]:

```
plt.figure(figsize = (15,10))

plt.subplot(2,2,1)
dendrogram(H_single)
plt.title("single")

plt.subplot(2,2,2)
dendrogram(H_complete)
plt.title ("complete")

plt.subplot(2,2,3)
dendrogram(H_average)
plt.title("average")

plt.show()
```



In [ ]:

```
# cut the dendogram
groups = fcluster(H_average, t = 27, criterion = "maxclust")
plt.figure(figsize = (10,7))
dendrogram(H_average)
plt.axhline(y = 27, color = 'r', linestyle = "--")
plt.show()
```



Density-based Clustering

```
In [ ]: df[['Number of Comments', 'sentiment_score_vader']].shape
```

```
Out[ ]: (3452, 2)
```

```
In [ ]: db =DBSCAN(eps = 0.1,min_samples =3).fit(df[['Number of Comments','sentiment_score_vader']])
```

```
In [ ]: # Evaluation
        sil_score_kmeans = silhouette_score(df[['Number of Comments', 'sentiment_score_vader']], kmeans.labels_)
        sil_score_hc = silhouette_score(df[['Number of Comments', 'sentiment_score_vader']], groups)
        sil_score_dbscan = silhouette_score(df[['Number of Comments', 'sentiment_score_vader']], db.labels_)
```

```
print(sil_score_kmeans)
print(sil_score_hc)
print(sil_score_dbscan)
```

```
0.6993406591002103
0.476129360170812
0.42662650847816064
```

Classification

In [ ]:

```
# Define Predictors and Target variable
```

In [ ]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import numpy as np

# Initialize TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(min_df=10, max_df=0.90)

# Convert phrases into lower case and apply TF-IDF vectorization
tfidf_dtm = tfidf_vectorizer.fit_transform(df['processed'].apply(lambda x: ' '.join(x) if isinstance(x, list) else x))

# Calculate sum of TF-IDF scores and get the indices of the top 40 terms
tfidf_sum_scores = tfidf_dtm.sum(axis=0).A1
tfidf_top_40_indices = tfidf_sum_scores.argsort()[-40:][::-1]

# Create the dataframe using only the columns for the top 40 terms
df_top_40_terms_tfidf = pd.DataFrame(tfidf_dtm[:, tfidf_top_40_indices].toarray(), columns=np.array(tfidf_vectorizer.get_feature_names_
df_top_40_terms_tfidf
```

Out[ ]:

	im	weight	eat	day	ive	like	week	get	feel	calori	...	back	month	make	help	anyon	gy
0	0.000000	0.226704	0.208797	0.100705	0.000000	0.031046	0.036923	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.0	0.041935	0.0	0.0000
1	0.000000	0.000000	0.000000	0.261549	0.000000	0.000000	0.143844	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.0000
2	0.049363	0.244339	0.065637	0.253256	0.062922	0.000000	0.208924	0.000000	0.000000	0.079543	...	0.000000	0.080181	0.0	0.079096	0.0	0.0000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.0000
4	0.094548	0.077999	0.125717	0.040423	0.080346	0.000000	0.177850	0.000000	0.000000	0.050785	...	0.000000	0.000000	0.0	0.000000	0.0	0.1136
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
3447	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.117483	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.0000
3448	0.000000	0.000000	0.088084	0.000000	0.084442	0.000000	0.093459	0.000000	0.000000	0.000000	...	0.109958	0.000000	0.0	0.106146	0.0	0.0000
3449	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.277998	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.1792
3450	0.000000	0.000000	0.070858	0.068351	0.000000	0.000000	0.075182	0.197518	0.000000	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.0000
3451	0.000000	0.097126	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.0	0.000000	0.0	0.0000

3452 rows × 40 columns

In [ ]:

```
X = df_top_40_terms_tfidf
Y = df['Decision']
```

Data Partition

In [ ]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size =0.2, random_state =42)
```

Decision Tree with pruning

In [ ]:

```
from sklearn.tree import DecisionTreeClassifier

# Initialize the DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=42)

# Train the classifier
clf.fit(X_train, Y_train)
```

Out[ ]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

In [ ]:

```
# Fine-tune cost-complexity alpha value
path = clf.cost_complexity_pruning_path(X_train, Y_train)
ccp_alphas = path.ccp_alphas[:-1] # Exclude the maximum value; without any split
```

In [ ]:

```
# create a table containing both error rate and ccp_alpha
# Obtain train and test scores using validation_curve
train_scores, test_scores = validation_curve (
    clf, X_train, Y_train, param_name="ccp_alpha", param_range=ccp_alphas, cv=5, scoring="accuracy"
)
train_error_rates = 1 - train_scores
test_error_rates = 1 - test_scores
# Display the results using ValidationCurveDisplay
display = ValidationCurveDisplay(
    param_name= 'ccp_alpha', param_range=ccp_alphas,
    train_scores=train_error_rates, test_scores=test_error_rates, score_name="Accuracy"
)
display.plot()
plt.scale('log')
pit.ylabel ('Error Rate')
plt.show()
```

In [ ]:

```
# Final the optimal alpha value with lowest error rates.
mean_test_error = np.mean(test_error_rates, axis=1)
std_test_error = np.std(test_error_rates, axis=1)  # Corrected variable name
df_ccp = pd.DataFrame({
    'ccp_alpha': ccp_alphas,
    'mean_test_score': mean_test_error,
    'std_test_score': std_test_error
})
pd.set_option('display.max_rows', None)
df_ccp.sort_values('mean_test_score').head()
```

Out[ ]:

	ccp_alpha	mean_test_score	std_test_score
299	0.002056	0.387181	0.016560
298	0.002053	0.387181	0.016560
300	0.002166	0.391165	0.016627
294	0.001826	0.392609	0.015251
295	0.001839	0.392972	0.015494

In [ ]:

```
# Prune the tree based on the optimal ccp (best pruned tree)
Pruned_clf = DecisionTreeClassifier(ccp_alpha=0.002056)  # Replace 0.002056 with your optimal ccp_alpha value
Pruned_clf.fit(X_train, Y_train)
```

Out[ ]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(ccp_alpha=0.002056)
```

In [ ]:

```python
# accuracy
# model evaluation predicted class (start here)
Y_pred = Pruned_clf.predict(X_test)
dt_accuracy =accuracy_score(Y_test, Y_pred)
dt_accuracy
```

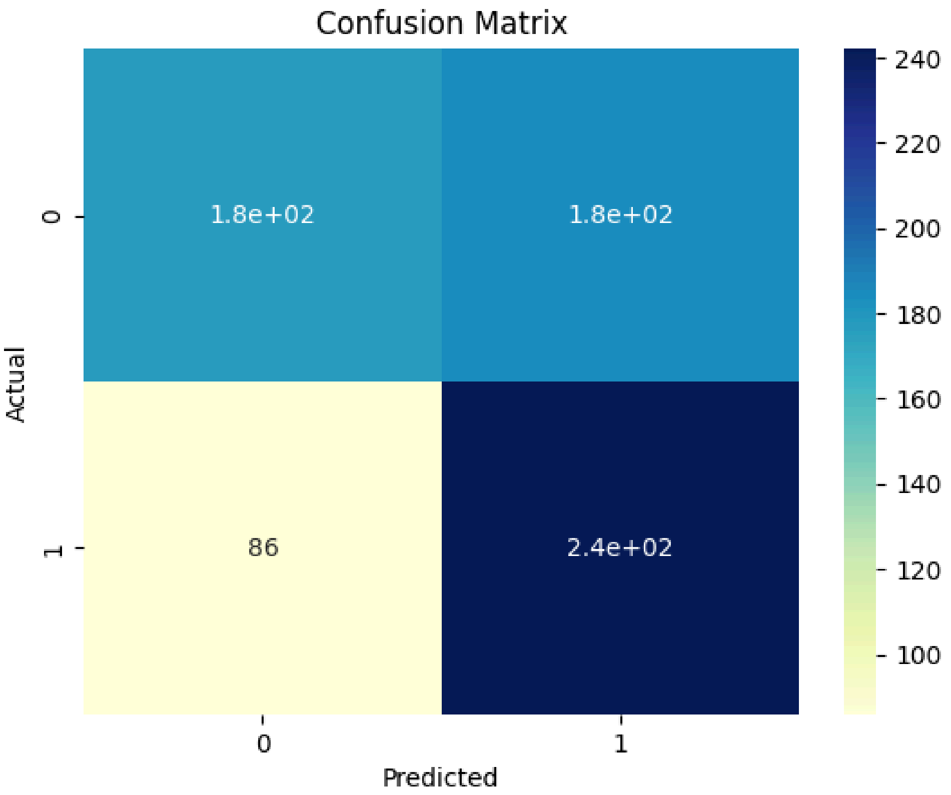
Out[ ]:

```
0.6078147612156295
```

In [ ]:

```python
# Confusion matrix
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
sns.heatmap(cm, annot=True, cmap="YlGnBu")  # Corrected colormap name
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

```
[[178 185]
 [ 86 242]]
```



Bagging classifier

```
In [ ]: tree_clf = DecisionTreeClassifier()
        bag_clf = BaggingClassifier(
            estimator = tree_clf,
            n_estimators = 100, # The number of base estimators in the ensemble
            bootstrap = True, # Whether samples are drawn with replacement. If False, sampling without replacement is performed.
            n_jobs = -1, # The number of jobs to run in parallel; -1 means using all processors
            random_state = 30)

        bag_clf.fit(X_train, Y_train)
```

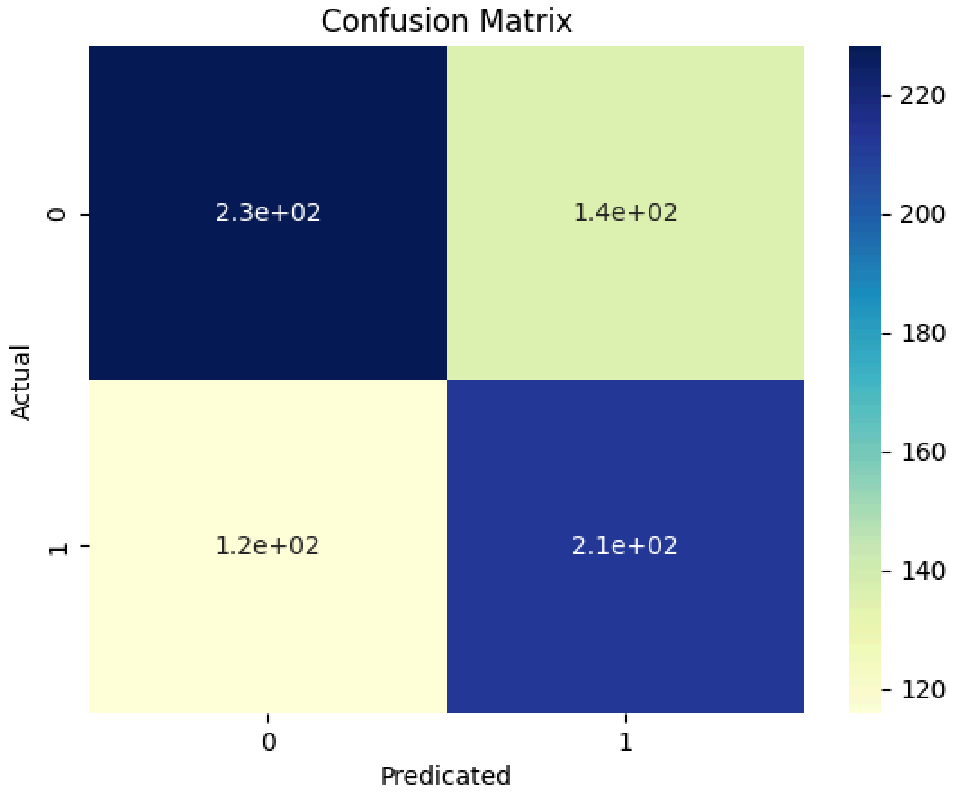
```
Out[ ]: ▸ BaggingClassifier
        ▸ estimator: DecisionTreeClassifier
            ▸ DecisionTreeClassifier
```

```
In [ ]: Y_pred = bag_clf.predict(X_test)
        bagging_accuracy = accuracy_score(Y_test, Y_pred)
        bagging_accuracy
```

```
Out[ ]: 0.6367583212735166
```

```
In [ ]: cm = confusion_matrix(Y_test, Y_pred)
        print(cm)
        sns.heatmap(cm, annot = True, cmap ="YlGnBu")
        plt.xlabel("Predicated")
        plt.ylabel("Actual")
        plt.title("Confusion Matrix")
        plt.show()
```

```
[[228 135]
 [116 212]]
```

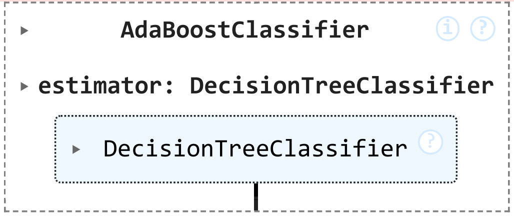


Boostraping Classifier

```
In [ ]: ada_clf = AdaBoostClassifier (
            estimator = tree_clf,
            n_estimators = 100, # The maximum number of estimators at which boosting is terminated.
            algorithm = "SAMME.R",
            learning_rate = 0.5, # Weight applied to each classifier at each boosting iteration.
            random_state = 30)

        ada_clf. fit(X_train, Y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_weight_boosting.py:519: FutureWarning: The SAMME.R algorithm (the default) i
s deprecated and will be removed in 1.6. Use the SAMME algorithm to circumvent this warning.
  warnings.warn(
```

Out[ ]:



AdaBoostClassifier

estimator: DecisionTreeClassifier

DecisionTreeClassifier

In [ ]:

```
Y_pred = ada_clf.predict(X_test)
boostraping_accuracy = accuracy_score(Y_test, Y_pred)
boostraping_accuracy
```

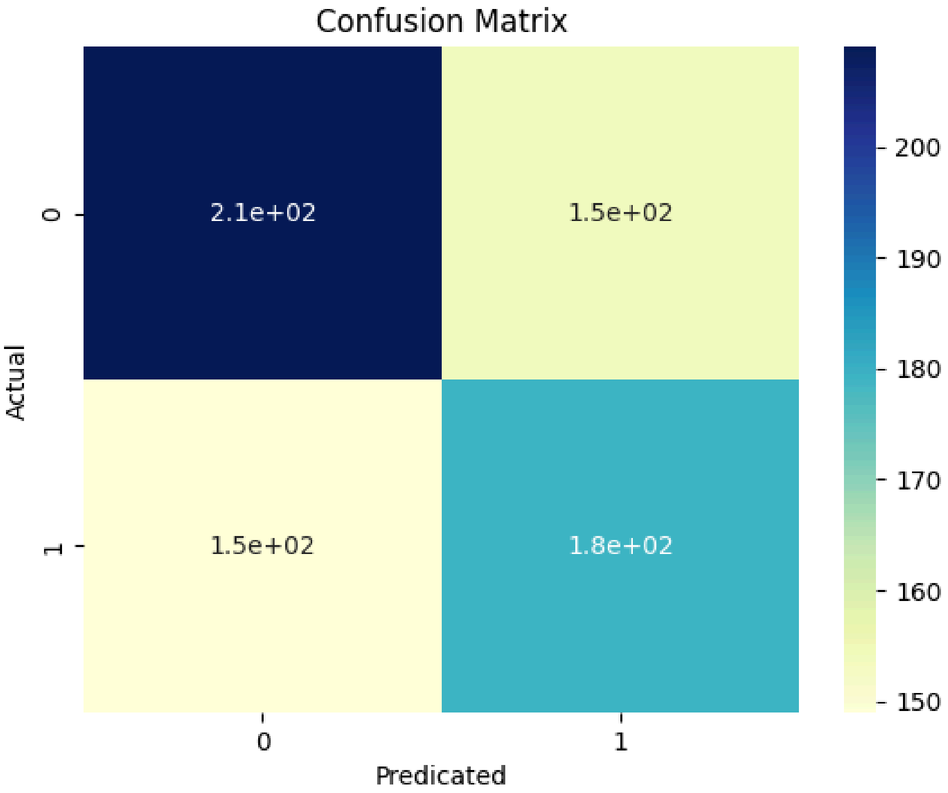
Out[ ]:

```
0.5615050651230101
```

In [ ]:

```
cm = confusion_matrix(Y_test, Y_pred)
print(cm)
sns.heatmap(cm, annot = True, cmap ="YlGnBu")
plt.xlabel("Predicated")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

```
[[209 154]
 [149 179]]
```



RandomForest Classifier

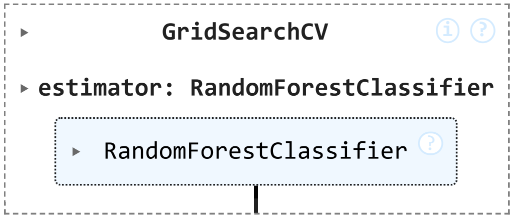
```
In [ ]: X.shape
Out[ ]: (3452, 40)
```

```
In [ ]: rf_clf = RandomForestClassifier(n_estimators=50, random_state=42, max_features=2, oob_score=True) # corrected argument name
```

```
In [ ]: from sklearn.model_selection import GridSearchCV

        param_grid = {'max_features': list(range(1, 41))}
        rf_grid_search = GridSearchCV(estimator=rf_clf, param_grid=param_grid, cv=5, n_jobs=-1, verbose=2)
        rf_grid_search.fit(X_train, Y_train)
```

```
Fitting 5 folds for each of 40 candidates, totalling 200 fits
```

Out[ ]:



In [ ]:

```
best_grid_rf = rf_grid_search.best_estimator_
print(best_grid_rf)
```

```
RandomForestClassifier(max_features=36, n_estimators=50, oob_score=True,
                       random_state=42)
```

In [ ]:

```
# Instantiate Random Forest Classifier
rf_c1f2 = RandomForestClassifier(n_estimators=50, random_state=42, max_features=36)
# Train classifier
rf_c1f2.fit(X_train, Y_train)
# Make predictions and evaluate
Y_pred_rf2 = rf_c1f2.predict(X_test)
RandomForestClassifier_accuracy = accuracy_score(Y_test, Y_pred_rf2)
RandomForestClassifier_accuracy
```

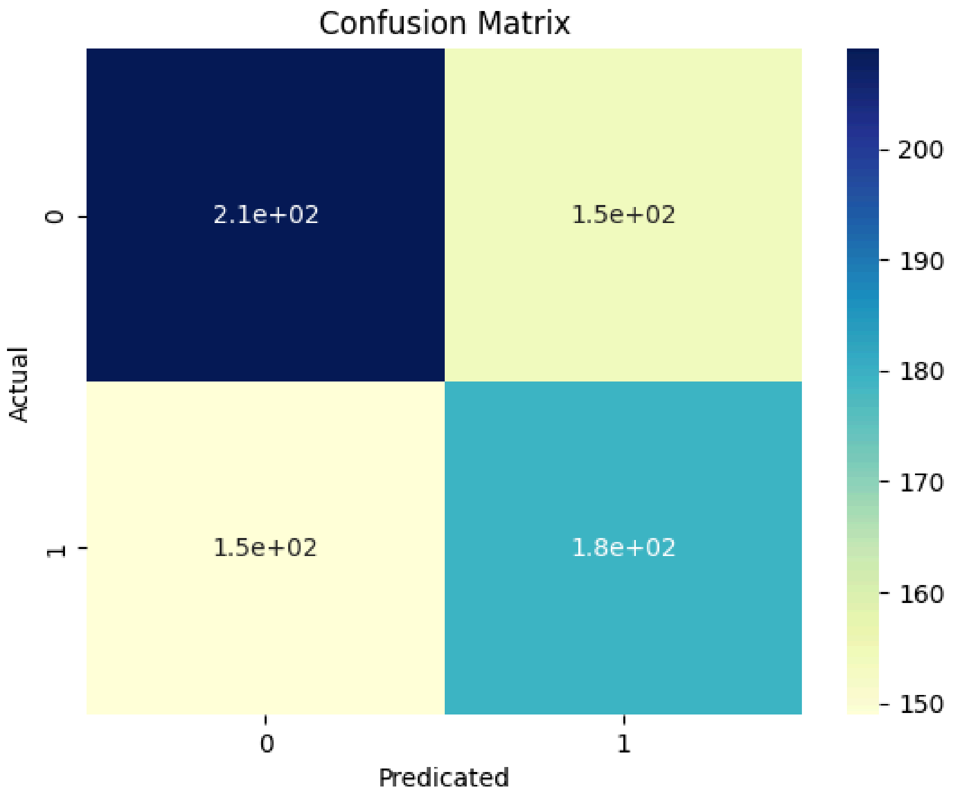
Out[ ]:

```
0.6121562952243126
```

In [ ]:

```
# 2. confusion matrix
cm = confusion_matrix (Y_test, Y_pred)
print(cm)
sns.heatmap(cm, annot = True, cmap = "YlGnBu")
plt.xlabel("Predicated" )
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

```
[[209 154]
 [149 179]]
```



In [ ]:

```
# Final Accuracy for the four classification models
# Final Accuracy for the four classification models
print('Decision Tree Classifier: ' + str(dt_accuracy))
print('Bagging Classifier: ' + str (bagging_accuracy))
print('Boostraping Classifier: ' +str (boostraping_accuracy))
print('RandomForest Classifier: ' +str (RandomForestClassifier_accuracy))
```

```
Decision Tree Classifier: 0.6078147612156295
Bagging Classifier: 0.6367583212735166
Boostraping Classifier: 0.5615050651230101
RandomForest Classifier: 0.6121562952243126
```