Home Assignment 4 SE 412/512 Max Points: 100
Due date Nov 07, 11:59 PM

Question 4. **(30 points)** Implement K-Means clustering algorithm for the dataset (**buddymove.csv**) provided with this assignment. You can run the algorithm three times for the values of K =3, K=4, and K=5 for each run (here K refers to # of clusters). A sample code file (**KMeans Clustering.py**) has been attached with this assignment. The description of data set is given below:

1. **Data Set Information:**
   This dataset was populated from destination reviews published by 249 reviewers of holidayiq.com till October 2014. Reviews falling in 6 categories among destinations across South India were considered and the count of reviews in each category for every reviewer (traveler) is captured. More information about this dataset is available at the following link: https://archive.ics.uci.edu/ml/datasets/BuddyMove+Data+Set

2. **Attribute Information:**
   Attribute 1 : Unique user id
   Attribute 2 : Number of reviews on stadiums, sports complex, etc.
   Attribute 3 : Number of reviews on religious institutions
   Attribute 4 : Number of reviews on beach, lake, river, etc.
   Attribute 5 : Number of reviews on theatres, exhibitions, etc.
   Attribute 6 : Number of reviews on malls, shopping places, etc.
   Attribute 7 : Number of reviews on parks, picnic spots, etc.

You might only need to consider Attributes 2 through 7, because User_id can be dropped. The idea is to cluster the Users that have similar characteristics of posting a # of reviews. You can implement the algorithm in any programming language you want. However, I highly recommend you use Python, R, or Matlab. Your program for question 4 should have the following:

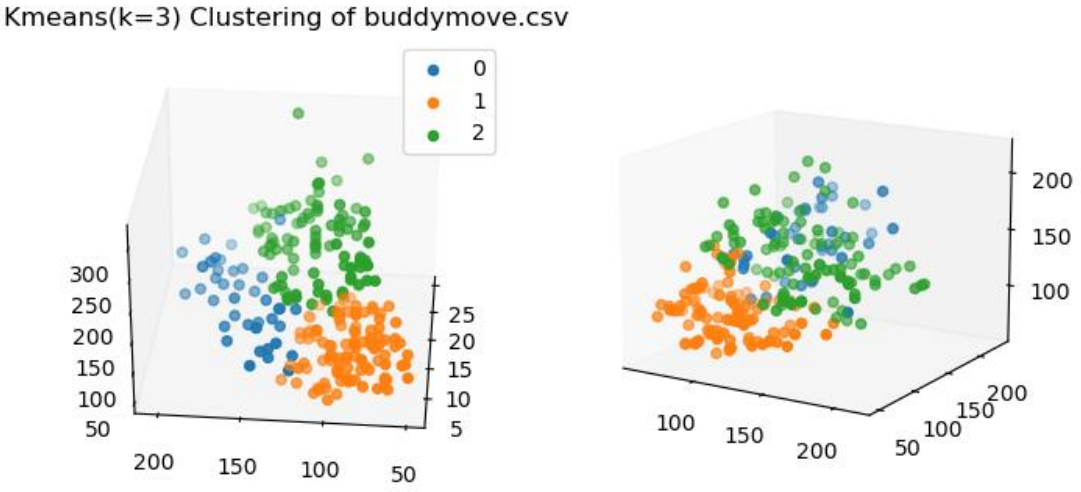
1. Attach a snapshot of each run of your program (for different values of K) that prints clustering labels for each run.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Jacob\Desktop\hw4> conda activate base
PS C:\Users\Jacob\Desktop\hw4> & C:/Users/Jacob/anaconda3/python.exe "c:/Users/Jacob/Desktop/hw4/KMean Clustering.py"
(249, 7)
   User Id  Sports  Religious  Nature  Theatre  Shopping  Picnic
0  User 1        2         77      79       69        68      95
1  User 2        2         62      76       76        69      68
2  User 3        2         50      97       87        50      75
3  User 4        2         68      77       95        76      61
4  User 5        2         98      54       59        95      86


Clustering buddymove.csv with K-Means(k=3)
```

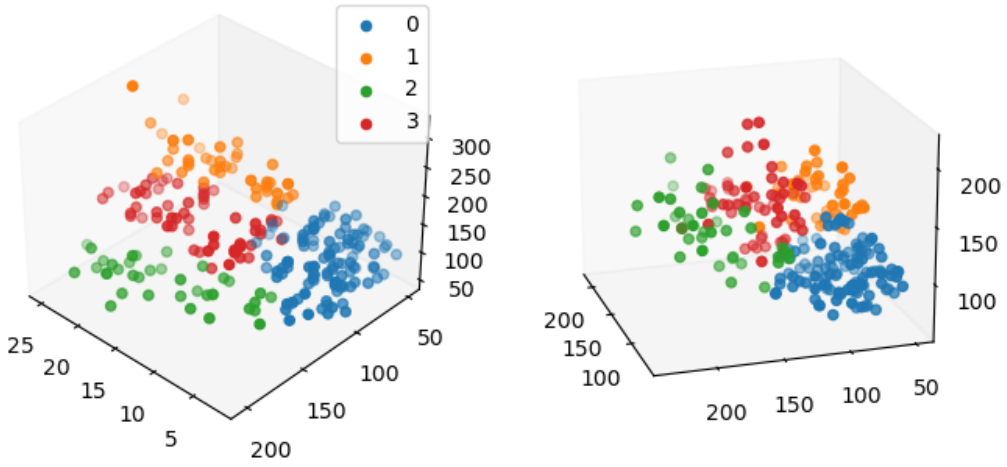
```
Clustering buddymove.csv with K-Means(k=3)
Unique labels:
[0 1 2]
Labels:
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 2 2
 2 2 2 2 0 0 2 2 2 2 2 2 0 1 2 2 2 2 0 2 1 2 1 2 2 0 2 2 1 0 1 2 0 1 0 2 1
 0 2 2 2 2 2 2 1 2 2 0 2 0 0 0 2 2 0 2 2 0 2 2 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 0 2 2 2 2 0 2 2 2 0 2 2 2 2 0 2 2 0 2 0 2 2 2 2 0 2 2 0 0 0 2 2 0 2 2 2 2
 2 2 0 2 2 0 2 2 2 0 2 2 2 2 0 1 2 2 0 2 2 2 2 2 2 2 2]
Centers:
[[ 15.30952381 159.45238095  85.30952381 108.83333333 179.07142857
  135.88095238]
 [  5.88181818  87.40909091 101.97272727  98.58181818  85.60909091
   94.51818182]
 [ 17.4742268  113.63917526 167.06185567 139.82474227 114.5257732
  143.05154639]]


Clustering buddymove.csv with K-Means(k=4)
Unique labels:
[0 1 2 3]
Labels:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0 0 0 0 2 0 0 2 0 0 0 0 0 0 3 3 3
 3 1 3 1 3 2 1 3 3 3 1 1 3 0 3 3 1 0 2 3 3 1 0 1 3 2 3 3 0 3 0 3 2 0 2 1 0
 2 1 1 1 1 3 3 0 3 1 2 3 2 3 2 1 1 2 3 1 2 3 1 3 1 1 3 2 1 3 1 2 1 3 3 2 1
 2 1 1 1 3 2 3 1 1 2 1 3 3 1 2 3 3 2 1 2 3 3 1 3 2 3 1 2 2 2 1 3 2 1 1 3 3
 3 3 2 1 3 2 3 3 3 2 3 1 3 1 2 0 3 3 2 1 1 1 3 1 3 3 3]
Centers:
[[  5.82727273  87.44545455 101.85454545  97.68181818  85.85454545
   94.48181818]
 [ 17.72727273  97.38636364 195.65909091 131.93181818  91.61363636
  147.22727273]
 [ 15.59459459 163.32432432  82.94594595 107.89189189 183.05405405
  137.89189189]
 [ 17.01724138 127.37931034 140.05172414 145.44827586 134.46551724
  138.05172414]]
```

```
Clustering buddymove.csv with K-Means(k=5)
Unique labels:
[0 1 2 3 4]
Labels:
[4 4 4 4 2 4 4 4 4 4 2 2 4 2 4 4 4 4 4 2 4 2 4 4 4 4 4 2 4 4 4 4 2 4 2 4 4
 4 4 2 4 4 4 4 4 4 2 2 4 4 3 4 4 4 4 2 4 4 2 4 2 4 4 4 2 4 4 4 2 4 2 4 4 4
 4 2 4 2 4 2 4 2 2 4 4 4 4 2 4 4 2 4 4 4 4 4 4 2 2 4 4 2 4 2 4 4 4 2 4 1 1 1
 1 0 1 0 1 3 0 1 1 1 0 0 1 4 1 1 0 4 3 1 1 0 4 0 1 3 1 1 4 1 4 1 3 4 3 0 4
 2 0 0 0 0 1 1 4 1 0 3 1 3 1 3 0 0 3 1 0 3 1 0 1 1 0 1 3 0 1 0 3 0 1 1 3 0
 3 0 0 0 1 3 1 0 0 3 0 1 1 0 3 1 1 3 0 3 1 1 0 1 3 1 0 3 3 3 0 1 3 0 0 1 1
 1 1 3 0 1 3 1 1 1 3 1 0 1 0 3 4 1 1 3 0 0 0 1 0 1 1 1]
Centers:
[[ 17.6744186   96.86046512 195.72093023 132.58139535  90.30232558
  147.44186047]
 [ 17.06779661 127.25423729 140.94915254 144.74576271 134.69491525
  138.05084746]
 [  6.09677419 116.         73.48387097  82.61290323 116.93548387
  106.09677419]
 [ 17.3        169.4        85.63333333 112.83333333 192.4
  143.16666667]
 [  5.93023256  81.20930233 109.60465116 102.22093023  79.30232558
   91.98837209]]
```

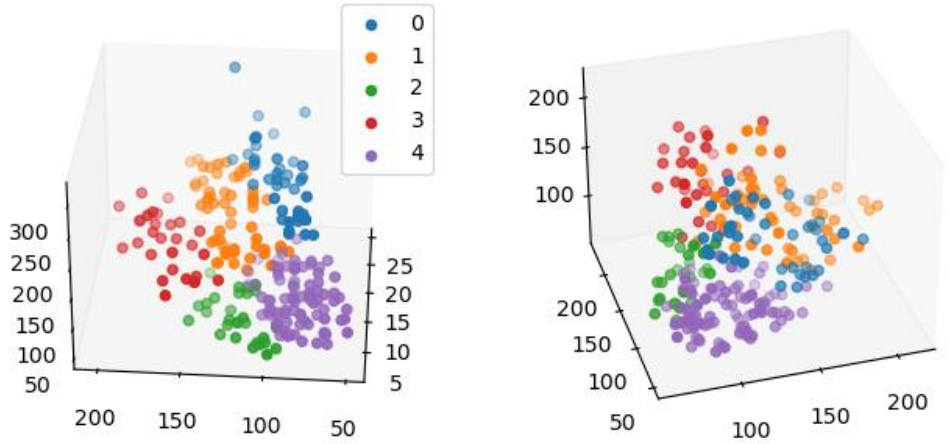
2. Your code should also print visualization of each cluster. Research on how to create a visualization for clusters.



Kmeans(k=4) Clustering of buddymove.csv



Kmeans(k=5) Clustering of buddymove.csv



3. Submit the code file along with your submission.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from sklearn.cluster import KMeans

##@Author: Jacob Hopkins
## 3d plot refernce https://likegeeks.com/3d-plotting-in-python/#Putting_legends

## read from csv file
dataset_filename = 'buddymove.csv'
data = pd.read_csv(dataset_filename)

##The following code displays the shape of data set
print(data.shape)
print(data.head())

#names of columns
f1label = 'Sports'
f2label = 'Religious'
f3label = 'Nature'
f4label = 'Theatre'
f5label = 'Shopping'
f6label = 'Picnic'

##Extracting each column
f1 = data[f1label]
f2 = data[f2label]
f3 = data[f3label]
f4 = data[f4label]
f5 = data[f5label]
f6 = data[f6label]

## Creating an array of data points
X = np.array(list(zip(f1,f2,f3,f4,f5,f6)))

## K-Means clustering algorithm with different parameters, 3-5 clusters

for k in range(3,6):
    print(f'\n\nClustering {dataset_filename} with K-Means(k={k})')

    kmeans = KMeans(n_clusters=k, random_state=0).fit(X)

```

```
47.     lables = kmeans.labels_
48.     unique_labels = np.unique(lables)
49.
50.     ## print cluster labels
51.     print("Unique labels: ")
52.     print(unique_labels)
53.
54.     print("Labels: ")
55.     print(lables)
56.
57.     print("Centers: ")
58.     print(kmeans.cluster_centers_)
59.
60.     ## visualize the clusters
61.     fig = plt.figure(figsize=(8,4))
62.
63.     ## plot of features 0 1 2
64.     ax = fig.add_subplot(121, projection='3d')
65.
66.     for l in unique_labels:
67.         ax.scatter(X[lables == l , 0] , X[lables == l , 1] , X[lables == l
    , 2], label = l)
68.
69.     ax.set_title(f'Kmeans(k={k}) Clustering of {dataset_filename}')
70.     ax.grid(False)
71.     ax.legend(loc="best")
72.
73.     ## plot of features 3, 4, 5
74.     ax2 = fig.add_subplot(122, projection='3d')
75.
76.     for l in unique_labels:
77.         ax2.scatter(X[lables == l , 3] , X[lables == l , 4] , X[lables ==
    l , 5], label = l)
78.
79.     ax2.grid(False)
80.     plt.show()
```