CSE 572: Homework 3

This notebook provides a template and starting code to implement the Homework 3 assignment.

To execute and make changes to this notebook, click File > Save a copy to save your own version in your Google Drive or Github. Read the step-by-step instructions below carefully. To execute the code, click on each cell below and press the SHIFT-ENTER keys simultaneously or by clicking the Play button.

When you finish executing all code/exercises, save your notebook then download a copy (.ipynb file). Submit the following **three** things:

1. a link to your Colab notebook,
2. the .ipynb file, and
3. a pdf of the executed notebook on Canvas.

To generate a pdf of the notebook, click File > Print > Save as PDF.

Real or spurious clusters?

An important question in assessing cluster validity is whether we are finding real patterns in structured data or finding patterns in noise or random data.

In this homework, you are given a dataset from an unknown source with unknown attributes. You are asked to cluster the data into 3 clusters using K-means clustering. Your goal is to evaluate whether the clusters you find in the dataset represent a valid clustering.

You can use any metric we have discussed in class or in the Data Mining textbook, but you must use a statistical test to evaluate validity of the clustering (refer to Lecture 19). Show all of your work and then answer the question in the final Question cell.

```
import pandas as pd
import numpy as np
np.random.seed(0)

data = pd.read_csv('https://docs.google.com/uc?export=download&id=1CjR6Q6nMN_2pTJJietr07mRjEYYSWR7U', header=None)
data.sample(10)
```

	0	1	2	3	4	5	6	
26	3.092368	5.402461	-8.133775	-0.294511	7.969525	-12.835632	-5.063480	7.273
86	-1.827800	6.874412	-6.193014	-4.061231	8.434069	-2.325613	-6.572138	2.662
2	5.118916	9.939231	-5.338687	-1.445761	8.098372	-5.761612	-12.977400	-0.516
55	3.714016	8.423915	3.827518	-4.131552	-2.399190	6.605014	-12.828675	-1.187
75	4.569833	11.060755	-6.875873	-2.357554	15.631446	-3.380032	-5.057848	3.413
93	-0.258834	9.663264	9.475254	-0.415799	-4.081936	0.444932	-5.416766	-4.677
16	-8.510063	-1.175450	-8.920655	-6.674918	7.775557	-0.555320	-5.426004	1.575
73	2.415988	6.377045	-5.826130	-2.328841	7.053033	-6.686211	-10.162534	1.710
54	-10.554498	-0.520517	-2.549914	-10.942731	5.034579	0.515082	-11.038140	-1.330
95	-8.379245	4.154273	-7.868257	-9.927669	6.662582	0.764915	-0.324643	2.055

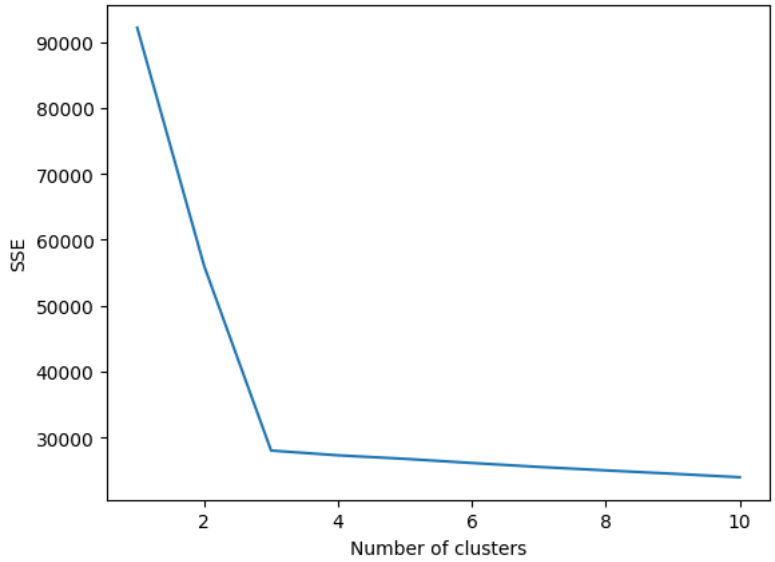
10 rows × 32 columns

```
# YOUR CODE HERE
random_seed = 0
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
kmeansWith3Clusters = KMeans(n_clusters=3, random_state=random_seed, init="random", n_init=10).fit(data)

print('SSE of Dataset: ', kmeansWith3Clusters.inertia_)
print('Silhouette score of Dataset:', silhouette_score(data, labels=kmeansWith3Clusters.labels_) )
```

```
SSE of Dataset:  28019.565034738327
Silhouette score of Dataset: 0.5012850888890975
```

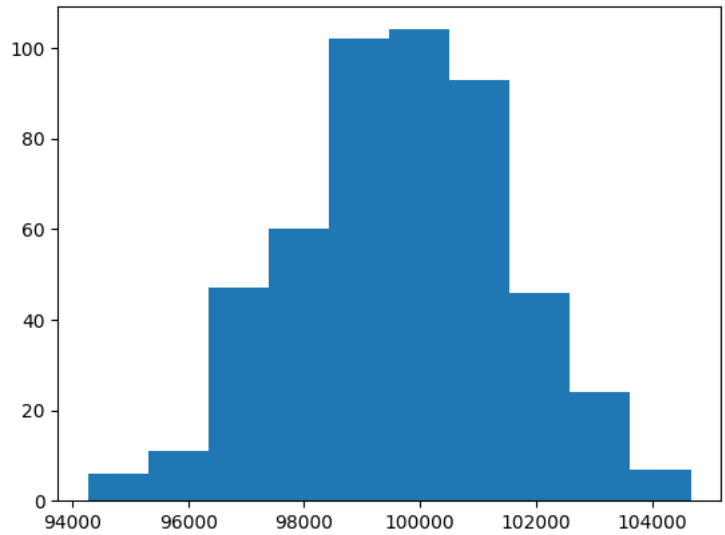
```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

sse = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=random_seed, init="random", n_init=10)
    kmeans.fit(data)
```

```
    sse.append(kmeans.inertia_)

plt.plot(range(1, 11), sse)
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.show()
```



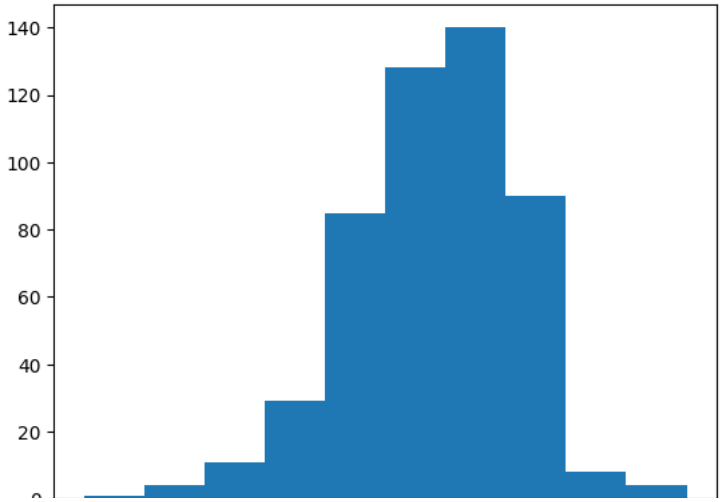

```
n = 500
rand_sse = []
silhouette_score_li = []
min_range = data.min()
max_range = data.max()

for i in range(n):
  np.random.seed(i)
  rand_data = pd.DataFrame()
  for i,col in enumerate(data.columns):
    rand_data[col]=list(np.random.uniform(min_range[i],max_range[i],size=data.shape[0]))
  km = KMeans(n_clusters=k, random_state=random_seed, init="random", n_init=10).fit(rand_data)
  rand_sse.append(km.inertia_)
  silhouette_score_li.append(silhouette_score(rand_data, labels=kmeans.labels_))


plt.hist(rand_sse)
```

```
(array([  6.,  11.,  47.,  60., 102., 104.,  93.,  46.,  24.,   7.]),
 array([ 94271.80250109,  95310.88180319,  96349.9611053 ,  97389.04040741,
         98428.11970952,  99467.19901163, 100506.27831374, 101545.35761584,
        102584.43691795, 103623.51622006, 104662.59552217]),
 <BarContainer object of 10 artists>)
```




```
plt.hist(silhouette_score_li)
```

```
(array([  1.,   4.,  11.,  29.,  85., 128., 140.,  90.,   8.,   4.]),
 array([-0.07970907, -0.07551286, -0.07131666, -0.06712045, -0.06292425,
        -0.05872804, -0.05453183, -0.05033563, -0.04613942, -0.04194322,
        -0.03774701]),
 <BarContainer object of 10 artists>)
```




```
print('SSE of random data: %f' % (np.mean(rand_sse)))

print('Mean Silhouette score of Test Data', np.mean(silhouette_score_li))
```

```
SSE of random data: 99674.481774
Mean Silhouette score of Test Data -0.05529387548306548
```

Question: Is your clustering result for the given dataset valid? Explain your answer. Answers must be justified using a statistical test with the chosen cluster validity metric.

Answer:

As we can see, from the above results, SSE of the original data set is 28019.565034738327 and Silhouette Score of the same is 0.5012850888890975, where as the SSE of randomly generated data is 99674.481774 and Silhouette score is -0.05529.

So, to conclude based on the statistical results above, SSE results vary very much from each other, also we know that, lesser the SSE score the better clustering, and similarly, Silhouette Score close to 1 meaning there is better clustering.

We can see that, we have don't have the best clusters but fat better than the randomly generated clusters. So, to conclude, our model performed very well forming clusters and they are REAL.