5505 Assignment 3: Clustering

Shiva Chakravarthy Gollapudi
Student id: 11468697

Clustering is used to measure the similarity or dissimilarity between data points, a distance measured is used.

One of the most commonly used distance functions is Euclidean.

$$\text{Euclidean: } \sqrt{\sum_{i=1}^{k}(xi - yi)2}$$

One of the most partition based clustering is K-means.

Data: ALS_TrainingData_2223.csv

Tools and Environment: Using python programming and Google Colab, I have explored the data and created Linear Regression models.

1. Load Libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.metrics import silhouette_score
```

2. Read Input:

```
df.head()
```

	ID	Age_mean	Albumin_max	Albumin_median	Albumin_min	Albumin_range	ALSFRS_slope	ALSFRS_Total_max	ALSFRS_Total_median	ALSFRS_Total_min	ALSFRS_Total_ra
0	1	65	57.0	40.5	38.0	0.066202	-0.965608	30	28.0	22	0.02
1	2	48	45.0	41.0	39.0	0.010453	-0.921717	37	33.0	21	0.02
2	3	38	50.0	47.0	45.0	0.008929	-0.914787	24	14.0	10	0.02
3	4	63	47.0	44.0	41.0	0.012111	-0.598361	30	29.0	24	0.01
4	5	63	47.0	45.5	42.0	0.008292	-0.444039	32	27.5	20	0.02

5 rows × 101 columns

There are 101 columns and 2223 rows in dataset.

3. Exploratory data analysis:

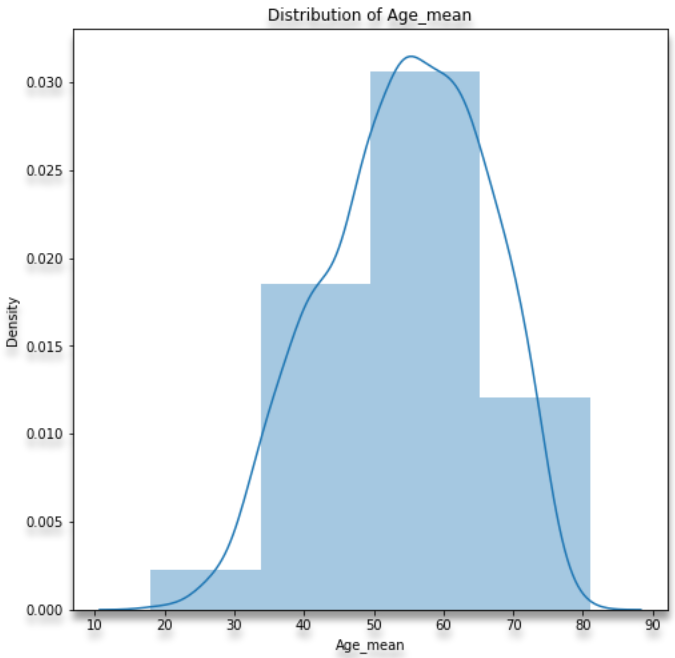
Finding the missing values: There are no missing/null values in the data.

```
[49] df.isnull().sum() #finding null values

     ID                  0
     Age_mean            0
     Albumin_max         0
     Albumin_median      0
     Albumin_min         0
                        ..
     trunk_range         0
     Urine.Ph_max        0
     Urine.Ph_median     0
     Urine.Ph_min        0
     cluster             0
     Length: 102, dtype: int64
```

There are no missing values.

Plotted distribution plot of Age_mean column. The data is not normally distribution.



Model 1: Building the K-mean (K =3) model for all the variables in the data.

K-mean: The goal of K-Means clustering is to divide n objects into k groups, with each object belonging to the cluster with the closest mean. The best number of clusters k that results in the most separation.

First, cluster the data into k groups, where k is a predefined.

Initially I have taken K value as 3.

```
[178] from sklearn.cluster import KMeans
      km = KMeans(n_clusters = 3, random_state=101) # Build a k-means clustering model

[179] km.fit(x) # Fit the model to data

      KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
             n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
             random_state=101, tol=0.0001, verbose=0)

[180] np.round(km.cluster_centers_) #Finding the centroids
```

finding the sum squared error for the Cluster (K=3)

```
[184] print('SSE of the based model: ', km.inertia_)

      SSE of the based model:  20529618202538.504
```

SSE (sum squared error): SSE means sum of the squared differences between each observation and mean of the cluster that it was assigned. Our SSE of the based model is too high. Increasing the number of clusters will reduce this SSE.

Identify optimum number of clusters:

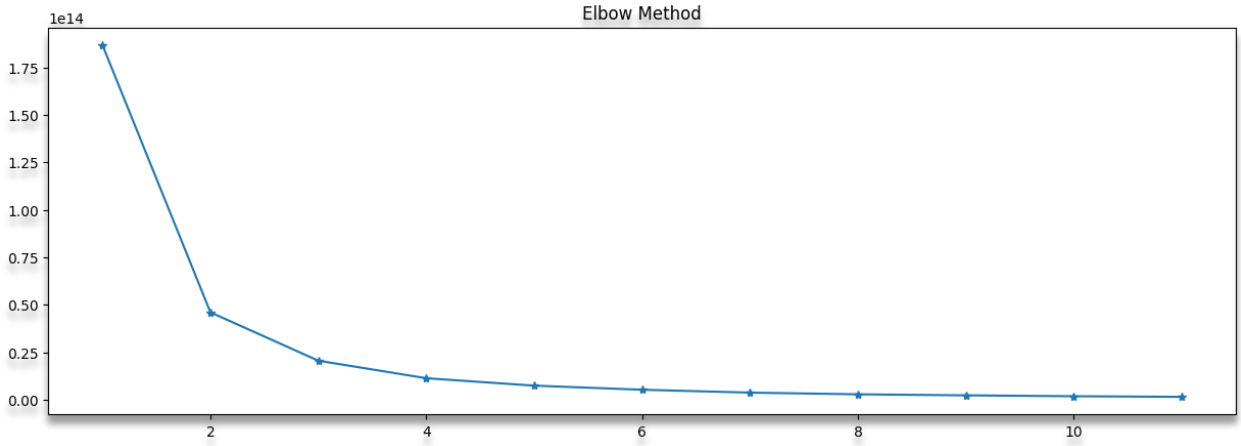
The low within sum of square will help to find the best k value.

```
wss = []

for i in range(1, 12):
    km = KMeans(n_clusters = i, random_state=101)
    km.fit(x)
    wss.append(km.inertia_)

plt.figure(figsize=(15,5), dpi = 100)
plt.plot(range(1, 12), wss,marker='*')
plt.title('Elbow Method')

plt.show();
```

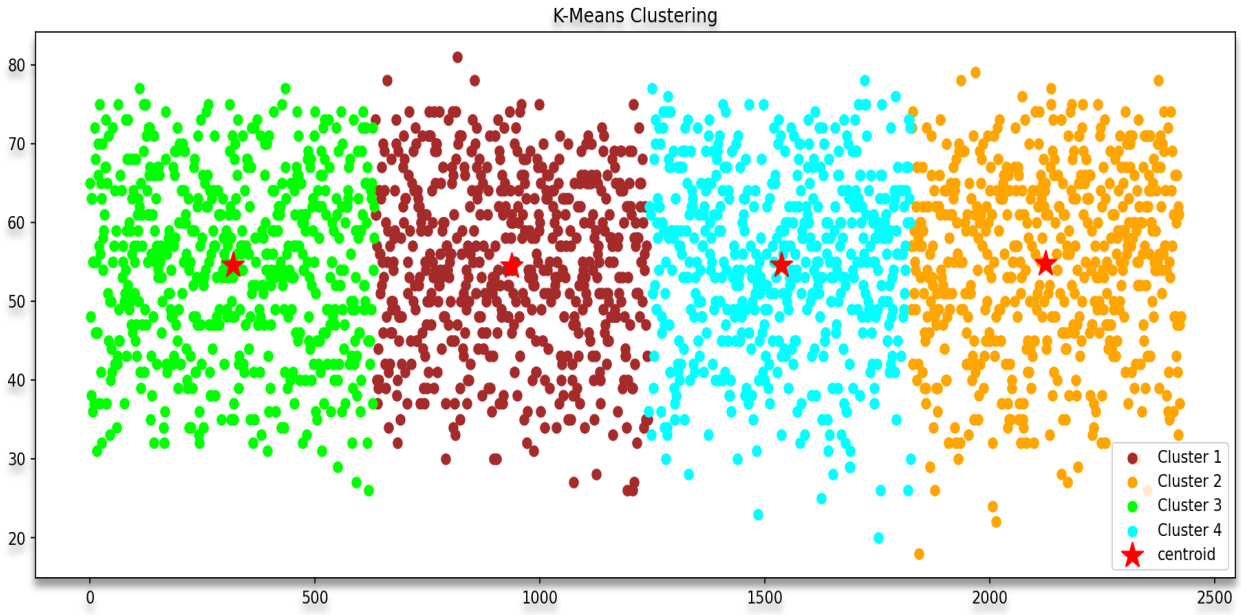


From the above diagram, we can conclude that Within sum of square is less for the cluster 4 and the rest all are negligible difference.

MODEL 2: Building the K-mean (k=4) model for all the variables in the data.

```
km4 = KMeans(n_clusters = 4, random_state=101)
km4.fit(x)
y = km4.fit_predict(x)
```

Plotted the scatter plot for the four clusters and their respective centroids.



The data has been divided into 4 clusters.

```
[39] pred = km4.predict(x)

     df['cluster'] = pred
     df['cluster'].value_counts() # Data distributed among four clusters
```

```
2    572
0    566
1    547
3    538
Name: cluster, dtype: int64
```

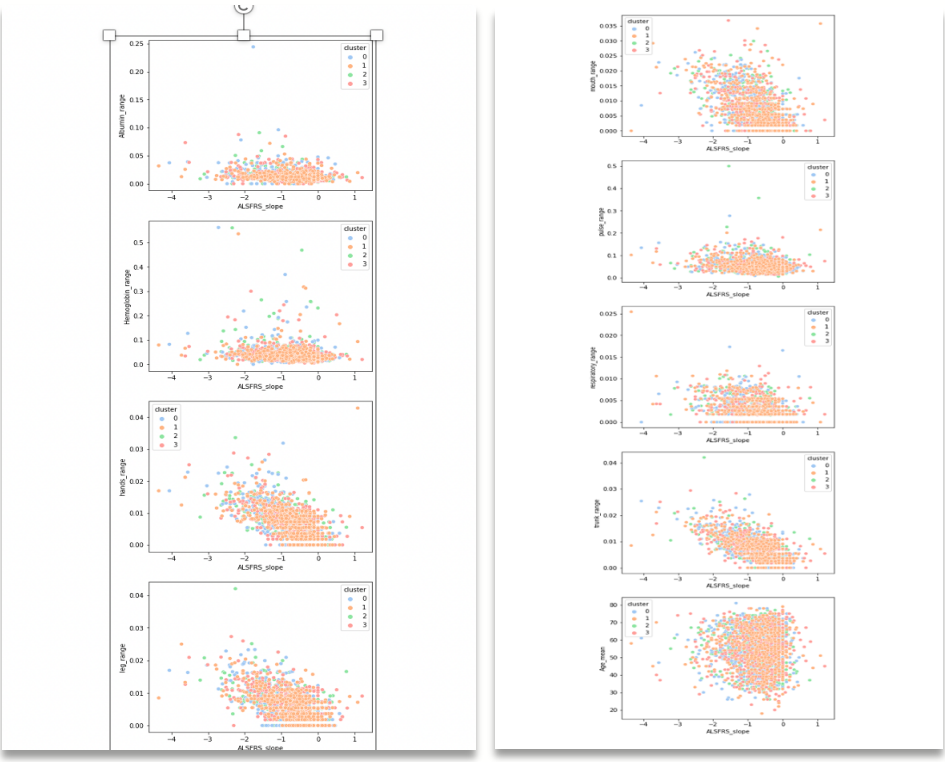
In first cluster, the number of records is 566, in the second 547. Third contains 572 and forth having 538 respectively.

Select some of the most important variables for visualizing:

```
Y = ['Albumin_range','Hemoglobin_range', 'hands_range',  'leg_range', 'mouth_range', 'pulse_range', 'respiratory_range', 'trunk_range', 'Age_mean']

# Plot 9 sub-scatter plots
fig, ax = plt.subplots(9,1,figsize=(6,50))
for i,y in enumerate(Y):
  sns.scatterplot ( 'ALSFRS_slope',y, hue = 'cluster', palette= 'pastel', data = df, ax=ax[i])

plt.show();
```



MODEL 3: Building the K-mean (k=4) model for all the two columns in the data.

Selected the two columns Age_mean and ALSFRS_slope

```
df_new=df2[["Age_mean","ALSFRS_slope"]]
df_new.head()
```

	Age_mean	ALSFRS_slope
0	65	-0.965608
1	48	-0.921717
2	38	-0.914787
3	63	-0.598361
4	63	-0.444039

Created a K-mean model with the k value as 3.

```
km = KMeans(n_clusters = 3, random_state=101) #For two columns, i have selected three clusters
```

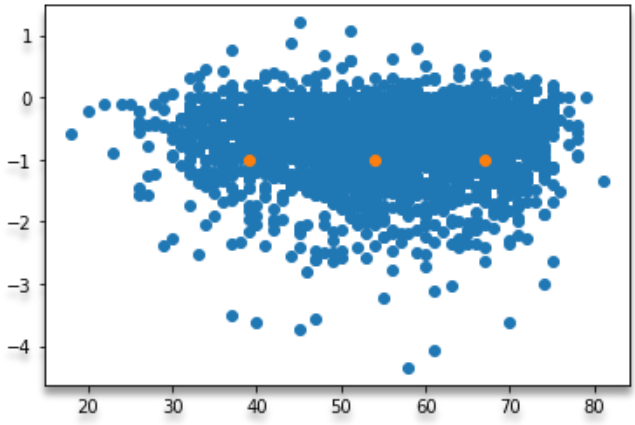
```
x = df_new.iloc[:,:].values
km.fit(x)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=101, tol=0.0001, verbose=0)
```

```
[161] y=np.round(km.cluster_centers_)
      y
```

Plotted the scatter plot for this model with the centroids.

```
plt.scatter(x[:,0],x[:,1])
plt.scatter(y[:,0],y[:,1]) #Plotted the scatter plot for the three clusters with centroids
```



Identify optimum number of clusters:

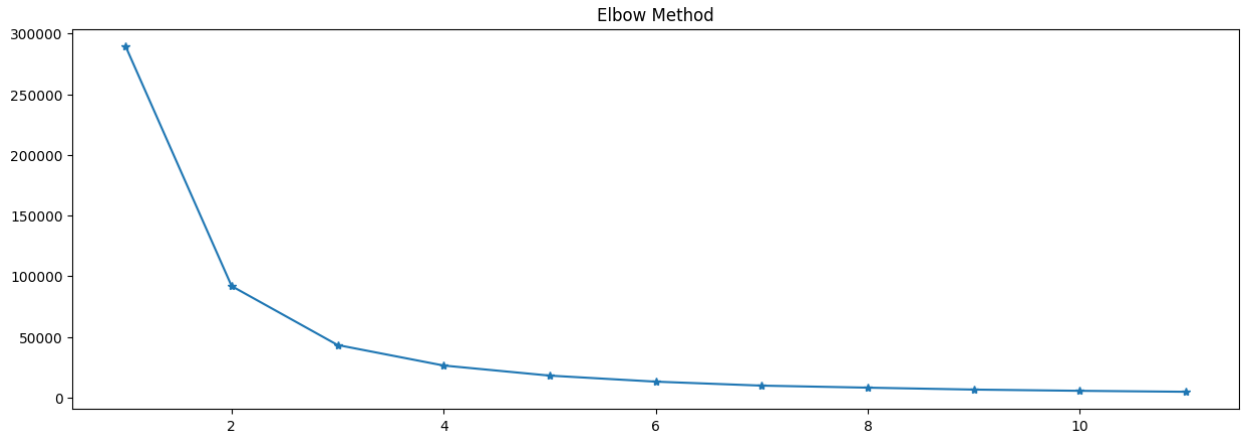
```
wss = []

for i in range(1, 12):
    km = KMeans(n_clusters = i, random_state=101)
    km.fit(x)
    wss.append(km.inertia_)

plt.figure(figsize=(15,5), dpi = 100)
plt.plot(range(1, 12), wss,marker='*')
plt.title('Elbow Method')

plt.show();
```

To identify the optimum number of clusters, we have used Elbow method.



From the above diagram, we can conclude that Within sum of square is less for the cluster 4 and the rest all are negligible difference.

MODEL 4: Building the K-mean (k=4) model for all the two variables selected.

```
[144] km = KMeans(n_clusters = 4, random_state=101)
      km.fit(x)
      y = km.fit_predict(x)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=101, tol=0.0001, verbose=0)
```

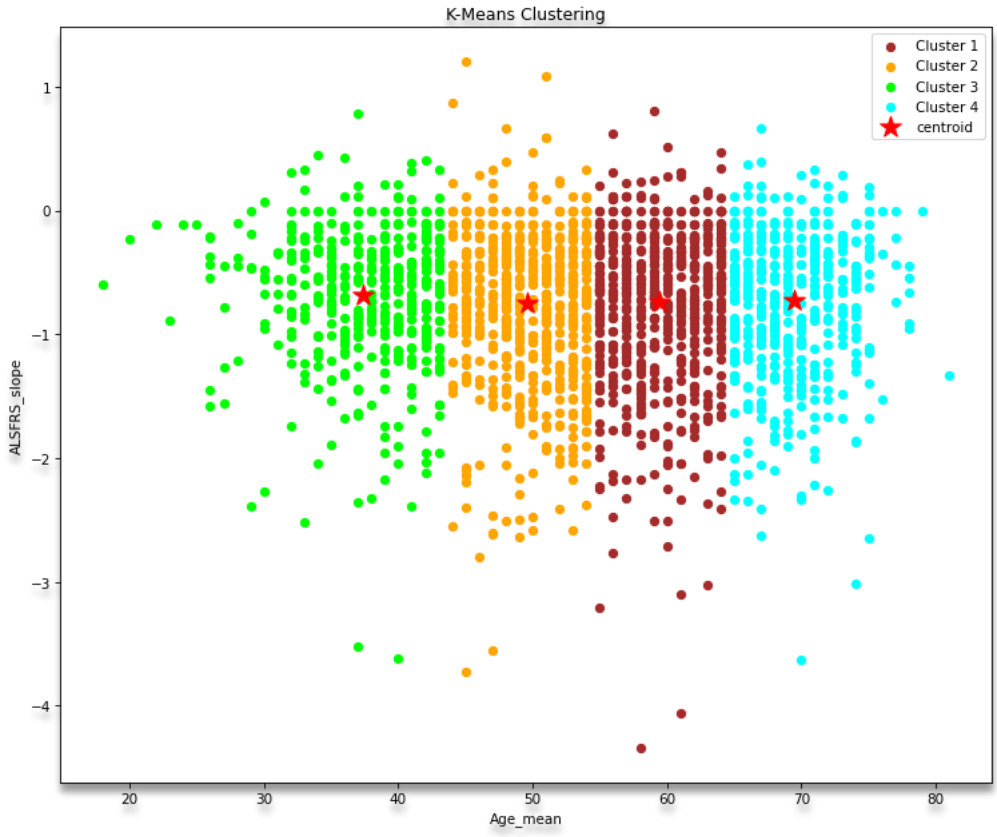
Plotted the scatter plot for this model with the centroids.

```
plt.figure(figsize=(12,10), dpi = 75)

plt.scatter(x[y == 0, 0], x[y == 0, 1], color = 'brown', label = 'Cluster 1')
plt.scatter(x[y == 1, 0], x[y == 1, 1], color = 'orange', label = 'Cluster 2')
plt.scatter(x[y == 2, 0], x[y == 2, 1], color = 'lime', label = 'Cluster 3')
plt.scatter(x[y == 3, 0], x[y == 3, 1], color = 'aqua', label = 'Cluster 4')

plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:,1],marker = '*', s = 250, c = 'red', label = 'centroid')

plt.ylabel('ALSFRS_slope')
plt.xlabel('Age_mean')
plt.title('K-Means Clustering')
plt.legend(loc = 'best')
plt.show();
```



Conclusion: Finally, we can conclude that the number of clusters 4 (K=4) is the best value to partition this dataset.