Import the necessary Libararies and dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv("/content/drive/MyDrive/DSC 630/als_data.csv")
data.head()
```

Out[2]:

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

5 rows × 101 columns

In [3]:
```
# Remove ID column
data = data.drop(['ID','SubjectID'],axis=1)
```

In [4]:
```
# Check the structure of the data

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2223 entries, 0 to 2222
Data columns (total 99 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Age_mean                         2223 non-null   int64  
 1   Albumin_max                      2223 non-null   float64
 2   Albumin_median                   2223 non-null   float64
 3   Albumin_min                      2223 non-null   float64
 4   Albumin_range                    2223 non-null   float64
 5   ALSFRS_slope                     2223 non-null   float64
 6   ALSFRS_Total_max                 2223 non-null   int64  
 7   ALSFRS_Total_median              2223 non-null   float64
 8   ALSFRS_Total_min                 2223 non-null   int64  
 9   ALSFRS_Total_range               2223 non-null   float64
 10  ALT.SGPT._max                    2223 non-null   float64
 11  ALT.SGPT._median                 2223 non-null   float64
 12  ALT.SGPT._min                    2223 non-null   float64
 13  ALT.SGPT._range                  2223 non-null   float64
 14  AST.SGOT._max                    2223 non-null   int64  
 15  AST.SGOT._median                 2223 non-null   float64
 16  AST.SGOT._min                    2223 non-null   float64
 17  AST.SGOT._range                  2223 non-null   float64
 18  Bicarbonate_max                  2223 non-null   float64
 19  Bicarbonate_median               2223 non-null   float64
 20  Bicarbonate_min                  2223 non-null   float64
 21  Bicarbonate_range                2223 non-null   float64
 22  Blood.Urea.Nitrogen..BUN._max    2223 non-null   float64
 23  Blood.Urea.Nitrogen..BUN._median 2223 non-null   float64
 24  Blood.Urea.Nitrogen..BUN._min    2223 non-null   float64
 25  Blood.Urea.Nitrogen..BUN._range  2223 non-null   float64
 26  bp_diastolic_max                 2223 non-null   int64  
 27  bp_diastolic_median              2223 non-null   float64
 28  bp_diastolic_min                 2223 non-null   int64  
 29  bp_diastolic_range               2223 non-null   float64
 30  bp_systolic_max                  2223 non-null   int64  
 31  bp_systolic_median               2223 non-null   float64
 32  bp_systolic_min                  2223 non-null   int64  
 33  bp_systolic_range                2223 non-null   float64
 34  Calcium_max                      2223 non-null   float64
 35  Calcium_median                   2223 non-null   float64
 36  Calcium_min                      2223 non-null   float64
 37  Calcium_range                    2223 non-null   float64
 38  Chloride_max                     2223 non-null   float64
 39  Chloride_median                  2223 non-null   float64
 40  Chloride_min                     2223 non-null   float64
 41  Chloride_range                   2223 non-null   float64
 42  Creatinine_max                   2223 non-null   float64
 43  Creatinine_median                2223 non-null   float64
 44  Creatinine_min                   2223 non-null   float64
 45  Creatinine_range                 2223 non-null   float64
 46  Gender_mean                      2223 non-null   int64  
 47  Glucose_max                      2223 non-null   float64
 48  Glucose_median                   2223 non-null   float64
 49  Glucose_min                      2223 non-null   float64
 50  Glucose_range                    2223 non-null   float64
 51  hands_max                        2223 non-null   int64  
 52  hands_median                     2223 non-null   float64
 53  hands_min                        2223 non-null   int64  
 54  hands_range                      2223 non-null   float64
 55  Hematocrit_max                   2223 non-null   float64
 56  Hematocrit_median                2223 non-null   float64
 57  Hematocrit_min                   2223 non-null   float64
 58  Hematocrit_range                 2223 non-null   float64
 59  Hemoglobin_max                   2223 non-null   float64
 60  Hemoglobin_median                2223 non-null   float64
 61  Hemoglobin_min                   2223 non-null   float64
 62  Hemoglobin_range                 2223 non-null   float64
 63  leg_max                          2223 non-null   int64  
 64  leg_median                       2223 non-null   float64
 65  leg_min                          2223 non-null   int64  
 66  leg_range                        2223 non-null   float64
```

```
 67  mouth_max                       2223 non-null   int64
 68  mouth_median                    2223 non-null   float64
 69  mouth_min                       2223 non-null   int64
 70  mouth_range                     2223 non-null   float64
 71  onset_delta_mean                2223 non-null   int64
 72  onset_site_mean                 2223 non-null   int64
 73  Platelets_max                   2223 non-null   int64
 74  Platelets_median                2223 non-null   float64
 75  Platelets_min                   2223 non-null   float64
 76  Potassium_max                   2223 non-null   float64
 77  Potassium_median                2223 non-null   float64
 78  Potassium_min                   2223 non-null   float64
 79  Potassium_range                 2223 non-null   float64
 80  pulse_max                       2223 non-null   int64
 81  pulse_median                    2223 non-null   float64
 82  pulse_min                       2223 non-null   int64
 83  pulse_range                     2223 non-null   float64
 84  respiratory_max                 2223 non-null   int64
 85  respiratory_median              2223 non-null   float64
 86  respiratory_min                 2223 non-null   int64
 87  respiratory_range               2223 non-null   float64
 88  Sodium_max                      2223 non-null   float64
 89  Sodium_median                   2223 non-null   float64
 90  Sodium_min                      2223 non-null   float64
 91  Sodium_range                    2223 non-null   float64
 92  trunk_max                       2223 non-null   int64
 93  trunk_median                    2223 non-null   float64
 94  trunk_min                       2223 non-null   int64
 95  trunk_range                     2223 non-null   float64
 96  Urine.Ph_max                    2223 non-null   float64
 97  Urine.Ph_median                 2223 non-null   float64
 98  Urine.Ph_min                    2223 non-null   float64
dtypes: float64(75), int64(24)
memory usage: 1.7 MB
```

The dataset contains 98 columns. We'll adopt randomforest regression for feature selection. From the literature, ALS condition is measured by ALSFRS_slope.

In [5]:
```
# import random forest libraries

from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectFromModel
```

In [6]:
```
# Create the features

X = data.drop('ALSFRS_slope',axis=1)
y = data['ALSFRS_slope']
```

In [7]:
```
# Create a randomforest regressor

clf = RandomForestRegressor(n_estimators=200,random_state=0,n_jobs=-1)

clf.fit(X,y)

# print the name of and gini importance of each feature

for feature in zip(X.columns,clf.feature_importances_):
  print(feature)
```

```
('Age_mean', 0.0037744688668357403)
('Albumin_max', 0.0019376217608390053)
('Albumin_median', 0.0018192338239591958)
('Albumin_min', 0.0017183887255982123)
('Albumin_range', 0.005264135674859937)
('ALSFRS_Total_max', 0.020532032092087688)
('ALSFRS_Total_median', 0.010089015881257025)
('ALSFRS_Total_min', 0.0033366382923384922)
('ALSFRS_Total_range', 0.6939651837415662)
('ALT.SGPT._max', 0.00295319904925661)
('ALT.SGPT._median', 0.0036660291425108862)
('ALT.SGPT._min', 0.003139578367509986)
('ALT.SGPT._range', 0.004751087090298099)
('AST.SGOT._max', 0.002432724048575397)
('AST.SGOT._median', 0.0032721341619729536)
('AST.SGOT._min', 0.002124354823806451)
('AST.SGOT._range', 0.0037533735963918334)
('Bicarbonate_max', 0.0021510402231933907)
('Bicarbonate_median', 0.0027435407955361552)
('Bicarbonate_min', 0.0021067745505395446)
('Bicarbonate_range', 0.0047492099309716556)
('Blood.Urea.Nitrogen..BUN._max', 0.0026961113635823705)
('Blood.Urea.Nitrogen..BUN._median', 0.0032930035472456976)
('Blood.Urea.Nitrogen..BUN._min', 0.0028739993049752555)
('Blood.Urea.Nitrogen..BUN._range', 0.0039664344459234505)
('bp_diastolic_max', 0.0023420990149761395)
('bp_diastolic_median', 0.002726738577044919)
('bp_diastolic_min', 0.0029040783062228683)
('bp_diastolic_range', 0.006009549936135036)
('bp_systolic_max', 0.003281655547487151)
('bp_systolic_median', 0.003194279148484597)
('bp_systolic_min', 0.0016222291502942263)
('bp_systolic_range', 0.004546157460037492)
('Calcium_max', 0.003842454772770578)
('Calcium_median', 0.003832219913324575)
('Calcium_min', 0.004523704904076454)
('Calcium_range', 0.00755071113843114)
('Chloride_max', 0.002122242028121051)
('Chloride_median', 0.0031226319569869943)
('Chloride_min', 0.0017390666810308728)
('Chloride_range', 0.00403117069097829)
('Creatinine_max', 0.004362647198038211)
('Creatinine_median', 0.001998213919830301)
('Creatinine_min', 0.002220724330725083)
('Creatinine_range', 0.003438747016236064)
('Gender_mean', 0.0001707104546061468)
('Glucose_max', 0.0027822005158054373)
('Glucose_median', 0.003835075427164478)
```

```
('Glucose_min', 0.0035936180224792136)
('Glucose_range', 0.0038133557410437568)
('hands_max', 0.0016508444766616486)
('hands_median', 0.003182895023346062)
('hands_min', 0.0009487691704029239)
('hands_range', 0.011313916283717002)
('Hematocrit_max', 0.004334419483766233)
('Hematocrit_median', 0.0024247411754134128)
('Hematocrit_min', 0.002364624207826169)
('Hematocrit_range', 0.002971208103657695)
('Hemoglobin_max', 0.003178924398053193)
('Hemoglobin_median', 0.0024527039269567202)
('Hemoglobin_min', 0.0024706885719664087)
('Hemoglobin_range', 0.003699016311182744)
('leg_max', 0.0007491980978650781)
('leg_median', 0.003994358323156322)
('leg_min', 0.0010704859743531627)
('leg_range', 0.005827842783422995)
('mouth_max', 0.0011821914670597994)
('mouth_median', 0.002699246434720386)
('mouth_min', 0.0015252558183068863)
('mouth_range', 0.00958253505828122)
('onset_delta_mean', 0.005944568629152332)
('onset_site_mean', 0.0003483322355555323)
('Platelets_max', 0.0032408888378353726)
('Platelets_median', 0.0030886501705636882)
('Platelets_min', 0.0029510453373391738)
('Potassium_max', 0.0022071866642747683)
('Potassium_median', 0.0032916545719424892)
('Potassium_min', 0.001884817224567828)
('Potassium_range', 0.004468731566231369)
('pulse_max', 0.0023282463562429375)
('pulse_median', 0.0035788196596888665)
('pulse_min', 0.0020645805877248903)
('pulse_range', 0.003644296754251333)
('respiratory_max', 0.00014079330156300239)
('respiratory_median', 0.0010243732421574847)
('respiratory_min', 0.0010046476800021657)
('respiratory_range', 0.010141287509838174)
('Sodium_max', 0.001824542348036824)
('Sodium_median', 0.0014565683457035544)
('Sodium_min', 0.0015975117448724337)
('Sodium_range', 0.003863864564495714)
('trunk_max', 0.0012038502370728525)
('trunk_median', 0.0038080348142703235)
('trunk_min', 0.0011710330365869936)
('trunk_range', 0.005156980371164798)
('Urine.Ph_max', 0.0011348931662468827)
('Urine.Ph_median', 0.001330407814205934)
('Urine.Ph_min', 0.00023876141966601707)
```

In [8]:
```
# Identify and select the most important features

sfm = SelectFromModel(clf,threshold=0.0035)
sfm.fit(X,y)
```

Out[8]:
```
SelectFromModel(estimator=RandomForestRegressor(n_estimators=200, n_jobs=-1,
                                                random_state=0),
                threshold=0.0035)
```

In [9]:
```
for feature_list in sfm.get_support(indices=True):
  feat_names = X.columns[feature_list]
  print(feat_names)
```

```
Age_mean
Albumin_range
ALSFRS_Total_median
ALSFRS_Total_range
ALT.SGPT._median
ALT.SGPT._range
AST.SGOT._range
Bicarbonate_range
Blood.Urea.Nitrogen..BUN._range
bp_diastolic_range
bp_systolic_range
Calcium_max
Calcium_median
Calcium_min
Calcium_range
Chloride_range
Creatinine_max
Glucose_median
Glucose_min
Glucose_range
hands_range
Hematocrit_max
Hemoglobin_range
leg_median
leg_range
mouth_range
onset_delta_mean
Potassium_range
pulse_median
pulse_range
respiratory_range
Sodium_range
trunk_median
trunk_range
```

We have reduced 98 features to 33 most important features. Clustering will be done with these features.

Create a dataframe of important features

In [10]:
```
df = data[['Age_mean','Albumin_range','ALSFRS_Total_median','ALSFRS_Total_range','ALT.SGPT._median','ALT.SGPT._range','AST.SGOT._ran
       'Bicarbonate_range','Blood.Urea.Nitrogen..BUN._range','bp_diastolic_range','bp_systolic_range','Calcium_max','Calcium_median',
       'Calcium_min','Calcium_range','Chloride_range','Creatinine_max','Glucose_median','Glucose_min','Glucose_range','hands_range',
       'Hemoglobin_range','leg_median','leg_range','mouth_range','onset_delta_mean','Potassium_range','pulse_median','pulse_range','r
       'Sodium_range','trunk_median','trunk_range']]
```

In [11]:
```
df.head()
```

Out[11]:

	Age_mean	Albumin_range	ALSFRS_Total_median	ALSFRS_Total_range	ALT.SGPT._median	ALT.SGPT._range	AST.SGOT._range	Bicarbonate_range	Blood.Ure
0	65	0.066202	28.0	0.021164	22.0	0.020906	0.027875	0.017422	
1	48	0.010453	33.0	0.028725	13.0	0.029617	0.029617	0.012195	
2	38	0.008929	14.0	0.025000	20.0	0.019643	0.010714	0.019643	
3	63	0.012111	29.0	0.014963	60.0	0.052369	0.032419	0.007481	
4	63	0.008292	27.5	0.020374	26.5	0.026534	0.024876	0.014925	

5 rows × 34 columns

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2223 entries, 0 to 2222
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Age_mean                        2223 non-null   int64
 1   Albumin_range                   2223 non-null   float64
 2   ALSFRS_Total_median             2223 non-null   float64
 3   ALSFRS_Total_range              2223 non-null   float64
 4   ALT.SGPT._median                2223 non-null   float64
 5   ALT.SGPT._range                 2223 non-null   float64
 6   AST.SGOT._range                 2223 non-null   float64
 7   Bicarbonate_range               2223 non-null   float64
 8   Blood.Urea.Nitrogen..BUN._range 2223 non-null   float64
 9   bp_diastolic_range              2223 non-null   float64
 10  bp_systolic_range               2223 non-null   float64
 11  Calcium_max                     2223 non-null   float64
 12  Calcium_median                  2223 non-null   float64
 13  Calcium_min                     2223 non-null   float64
 14  Calcium_range                   2223 non-null   float64
 15  Chloride_range                  2223 non-null   float64
 16  Creatinine_max                  2223 non-null   float64
 17  Glucose_median                  2223 non-null   float64
 18  Glucose_min                     2223 non-null   float64
 19  Glucose_range                   2223 non-null   float64
 20  hands_range                     2223 non-null   float64
 21  Hematocrit_max                  2223 non-null   float64
 22  Hemoglobin_range                2223 non-null   float64
 23  leg_median                      2223 non-null   float64
 24  leg_range                       2223 non-null   float64
 25  mouth_range                     2223 non-null   float64
 26  onset_delta_mean                2223 non-null   int64
 27  Potassium_range                 2223 non-null   float64
 28  pulse_median                    2223 non-null   float64
 29  pulse_range                     2223 non-null   float64
 30  respiratory_range               2223 non-null   float64
 31  Sodium_range                    2223 non-null   float64
 32  trunk_median                    2223 non-null   float64
 33  trunk_range                     2223 non-null   float64
dtypes: float64(32), int64(2)
memory usage: 590.6 KB
```

Apply a Standard Scalar to the data

In [13]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn import cluster
from sklearn import metrics
```

In [14]:
```
cols = df.columns
ms = MinMaxScaler()
df = ms.fit_transform(df)
```

In [15]:
```
df = pd.DataFrame(df,columns=[cols])
df.head()
```

Out[15]:

	Age_mean	Albumin_range	ALSFRS_Total_median	ALSFRS_Total_range	ALT.SGPT._median	ALT.SGPT._range	AST.SGOT._range	Bicarbonate_range	Blood.Ure
0	0.746032	0.271429	0.680000	0.179894	0.075676	0.007629	0.014543	0.081301	
1	0.476190	0.042857	0.813333	0.244165	0.027027	0.011288	0.015452	0.056911	
2	0.317460	0.036607	0.306667	0.212500	0.064865	0.007098	0.005590	0.091667	
3	0.714286	0.049656	0.706667	0.127182	0.281081	0.020846	0.016914	0.034913	
4	0.714286	0.033997	0.666667	0.173175	0.100000	0.009993	0.012979	0.069652	

5 rows × 34 columns

Create a Plot the cluster Silhouette score versu sthe number of clusters in a K-means cluster

In [16]:
```
krange = list(range(2,11))
avg_silhouettes = []
for n in krange:
  model = cluster.KMeans(n_clusters=n,random_state=10)
  cluster_assignments = model.fit_predict(df)
  silhoutte_avg = metrics.silhouette_score(df,cluster_assignments)
  avg_silhouettes.append(silhoutte_avg)
```

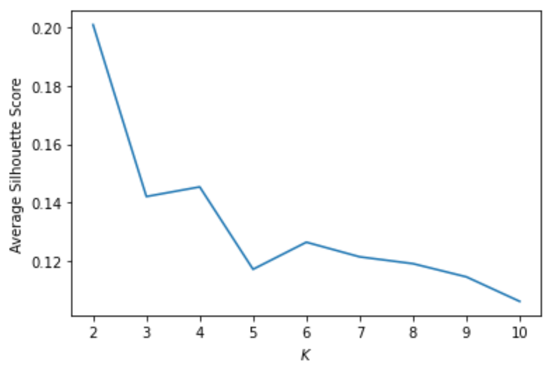
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
```

```
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
```

Plot the result of krange against avg_silhouette

In [17]:

```
plt.plot(krange,avg_silhouettes)
plt.xlabel("$K$")
plt.ylabel("Average Silhouette Score")
plt.show()
```



The above plot seem to suggest k = 2. since the highest average silhouette score was at its highest (0.20) when k =2.

Fit a K-means model to the data with k = 2

In [19]:

```
from sklearn.cluster import KMeans
km_2 = KMeans(n_clusters=2,random_state=42)
km_2.fit(df)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
```

Out[19]:

```
KMeans(n_clusters=2, random_state=42)
```

In [27]:

```
df['cluster'] = km_2.labels_
km2 = pd.DataFrame(km_2.labels_,columns=['cluster'])
km2.head()
```

Out[27]:

	cluster
0	0
1	0
2	1
3	0
4	1

Fitting a PCA

In [24]:

```
from sklearn.decomposition import PCA
```

In [30]:

```
pca = PCA(n_components=2).fit(df)
pca_trans = pca.transform(df)
pca_trans_df = pd.DataFrame(pca_trans,columns=['pca1','pca2'])

# concatenate km2 with pca_trans_df

km2 = pd.concat([km2,pca_trans_df],axis=1)
km2.head()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:1692: FutureWarning: Feature names only support names that are al
l strings. Got feature names with dtypes: ['tuple']. An error will be raised in 1.2.
  FutureWarning,
```

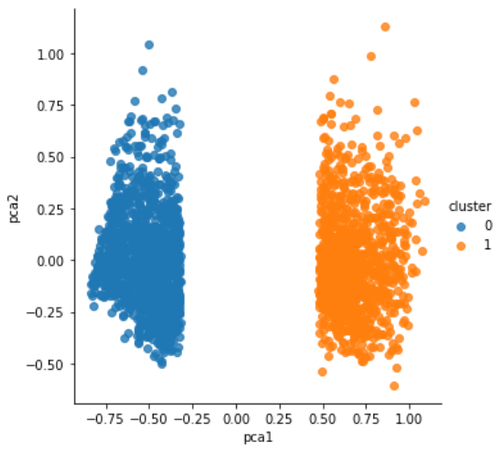
Out[30]:

cluster	pca1	pca2

	cluster	pca1	pca2
0	0	-0.560863	0.162707
1	0	-0.646207	0.207739
2	1	0.820034	-0.100761
3	0	-0.382765	-0.217366
4	1	0.576030	-0.288326

Plot the PCA

In [31]:
```
fig = sns.lmplot(x='pca1',y='pca2',data=km2,hue='cluster',fit_reg=False)
plt.show()
```



Summary of Results

This project utilized the Silhouette chart to select the number of optimal clusters in our dataframe. The dataframe consists of 101 features but 30 of the features were selected based on random forest regressor selection method with a threshold of 0.003.

Using the PCA to compress the dimension of the data, 2 clear groups were observed from the PCA visualization.

In [ ]:
```
```