baisravanhc

Assignment 4

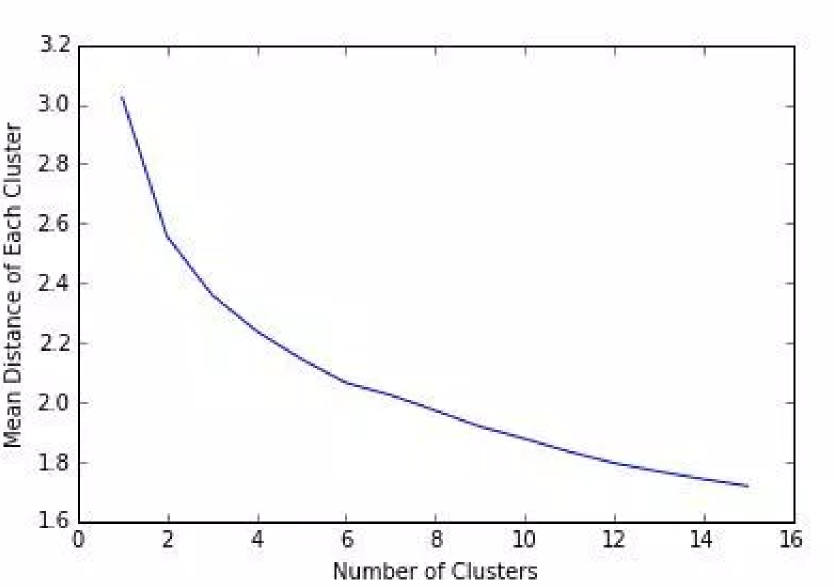
26 Friday Feb 2016

POSTED BY BAISRAVAN IN UNCATEGORIZED

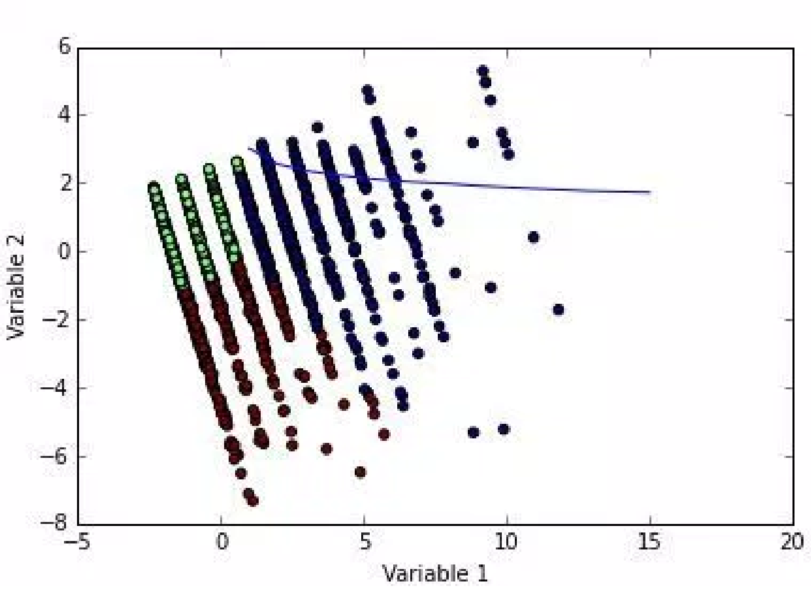
≈ LEAVE A COMMENT

In this assignment, I am clustering data with clustering variables: marijuana use, alcohol problem, deviant behavior scale, violent behavior scale, depression scale, self esteem scale, school connectedness scale and parental presence scale.

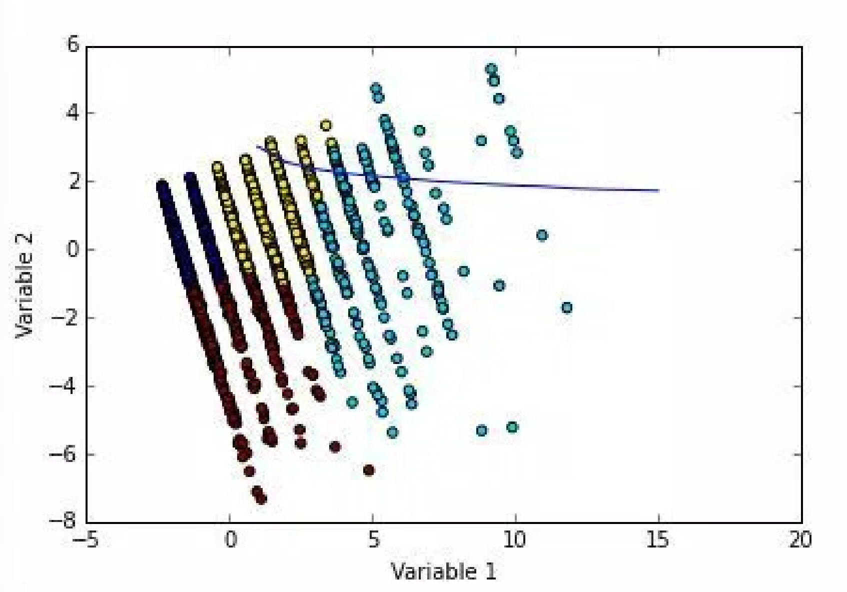
First, the data was processed and the graph showing the mean distance of the clusters with the number of clusters (elbow plot) was plotted. The plot is shown below:



Then I chose 3 clusters and the result is shown below:

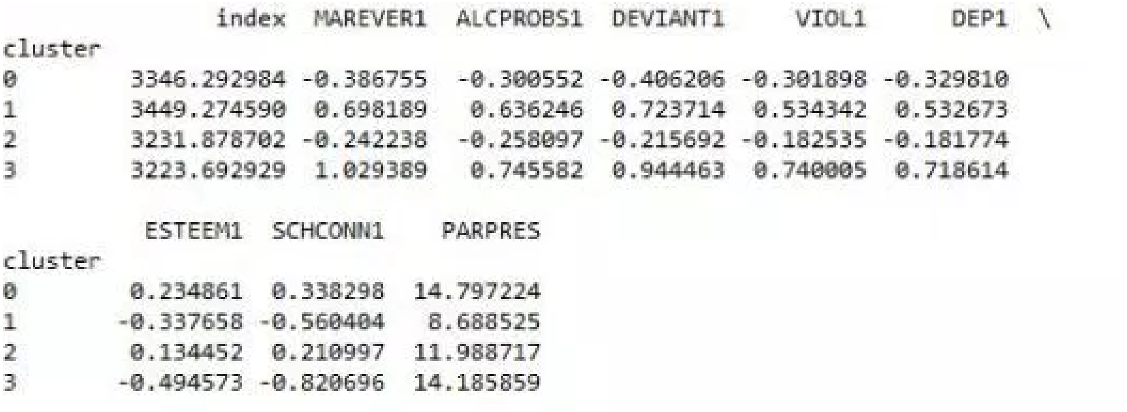


It seemed like another cluster would be beneficial:



Clearly 4 clusters seemed to classify the data better (It was seen that 5 clusters was not doing a good jobs).

The 4 clusters looks like this:



```
              index  MAREVER1  ALCPROBS1  DEVIANT1     VIOL1      DEP1  \
cluster
0       3346.292984 -0.386755  -0.300552 -0.406206 -0.301898 -0.329810
1       3449.274590  0.698189   0.636246  0.723714  0.534342  0.532673
2       3231.878702 -0.242238  -0.258097 -0.215692 -0.182535 -0.181774
3       3223.692929  1.029389   0.745582  0.944463  0.740005  0.718614

          ESTEEM1  SCHCONN1    PARPRES
cluster
0        0.234861  0.338298  14.797224
1       -0.337658 -0.560404   8.688525
2        0.134452  0.210997  11.988717
3       -0.494573 -0.820696  14.185859
```

It can be seen that 0 and cluster 2 were more or less similar in nature while cluster 1 and 3 showed higher likelihood of alcohol problem, violent and deviant behavior, depression, etc. These were low for the clusters 0 and 2.

CODE:

```
Spyder Editor

This is a temporary script file.
“””

import numpy as np
from pandas import DataFrame
import pandas as pd
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn import preprocessing
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
from sklearn.decomposition import PCA
data = pd.read_csv(‘tree_addhealth.csv’)

data.columns = map(str.upper,data.columns)

data = data.dropna()

cluster_var = data[[‘MAREVER1′,’ALCPROBS1′,’DEVIANT1′,’VIOL1’,
‘DEP1′,’ESTEEM1′,’SCHCONN1′,’PARPRES’]]

cluster_var.columns = map(str.upper,cluster_var.columns)
cluster_var.describe()
clusters = cluster_var.copy()
for i in range(0,len(cluster_var.columns)-1):
clusters[cluster_var.columns[i]] =
preprocessing.scale(clusters[cluster_var.columns[i]].astype(‘float64’))

cluster_train,cluster_test = train_test_split(clusters,test_size = 0.4,random_state = 123)

cluster_range = range(1,16)
mean_dist = []

for k in cluster_range:
model= KMeans(n_clusters = k)
model.fit(cluster_train)
mean_dist.append(sum(np.min(cdist(cluster_train, model.cluster_centers_, ‘euclidean’),
axis=1))
/ cluster_train.shape[0])

plt.plot(cluster_range,mean_dist)
plt.xlabel(‘Number of Clusters’)
plt.ylabel(‘Mean Distance of Each Cluster’)
```

```
model4 = KMeans(n_clusters = 3)
model4.fit(cluster_train)

pca2 = PCA(2)
plot_column = pca2.fit_transform(cluster_train)
plt.scatter(x = plot_column[:,0],y=plot_column[:,1], c=model4.labels_,)
plt.xlabel(‘Variable 1’)
plt.ylabel(‘Variable 2’)

cluster_train.reset_index(level=0, inplace=True)
cluster_list=list(cluster_train[‘index’])

labels=list(model4.labels_)

newlist=dict(zip(cluster_list, labels))
newclus=DataFrame.from_dict(newlist, orient=’index’)
newclus.columns = [‘cluster’]
newclus.reset_index(level=0, inplace=True)
merged_train=pd.merge(cluster_train, newclus, on=’index’)
merged_train.head(n=100)
merged_train.cluster.value_counts()

clustergrp = merged_train.groupby(‘cluster’).mean()
```