MT18052_Test1

September 6, 2019

In [45]:
```
import pandas as pd
import csv
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import matplotlib
from sklearn.manifold import TSNE
import warnings
warnings.filterwarnings("ignore")
svd = TruncatedSVD(n_components=2)

# matplotlib.use('Agg')
```

In [46]:
```
data = []
labels = []
with open('iris.data') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
#         data.append(row[0:4])
        labels.append(row[4])
        tmp = np.array(row[0:4]).astype(float)
        data.append(tmp)
data = np.array(data)
labels = np.array (labels)
```

In [47]:
```
print (len(data))
```

```
150
```

In [48]:
```
def plotgraph(resultlist,title=""):
    '''
    fig = plt.figure()
    ax = fig.gca(projection='3d')
    '''
#     plt.figure()
    for i in sorted(resultlist.keys()):
        '''
```

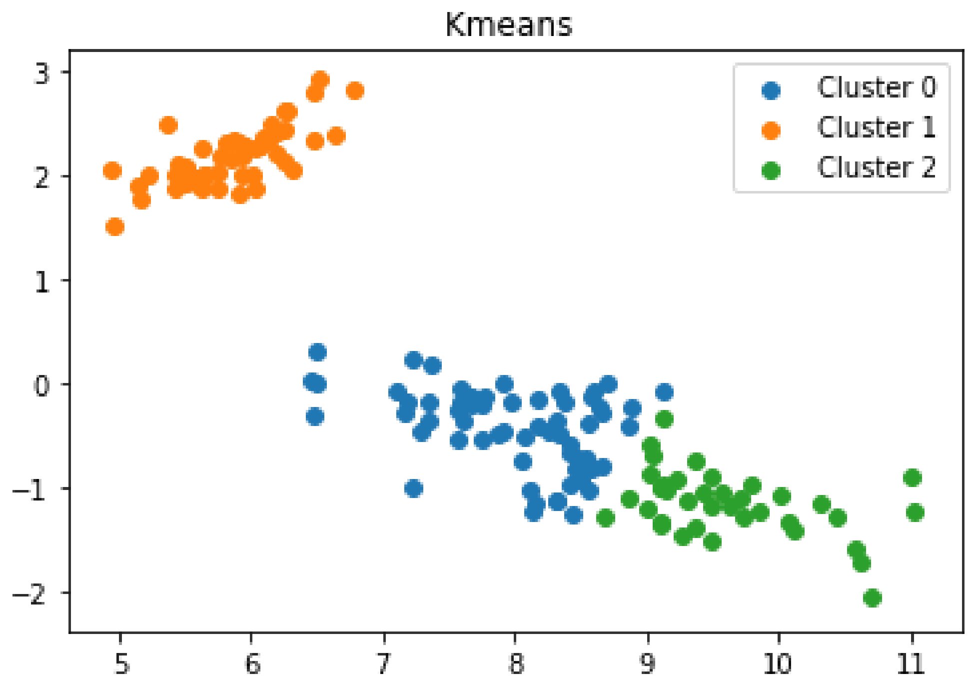
```
                x = resultlist[i].T[0]
                y = resultlist[i].T[1]
                z = resultlist[i].T[2]

                ax.scatter(x,y,z,label='Cluster ' + str(i))
                '''
                plt.scatter(resultlist[i].T[0],resultlist[i].T[1],label='Cluster ' + str(i))
                #plt.show()
            plt.legend(loc='best')
            plt.title(title)
            plt.show()

In [51]: def completefun(datapoints,numclusters=3):
         #     print (datapoints)
            kmeans = KMeans(n_clusters=numclusters, random_state=0).fit(datapoints)
         #     svd = TruncatedSVD(n_components=2)
            tnsepoints = svd.fit_transform(datapoints)
            clusters = {}
            predicted_labels = []
            for i in range(len(kmeans.labels_)):
                predicted_labels.append(kmeans.labels_[i])
                if(kmeans.labels_[i] not in clusters):
                    clusters[kmeans.labels_[i]] = []
                clusters[kmeans.labels_[i]].append(i)
            for i in clusters.keys():
                clusters[i] = tnsepoints[clusters[i]]
         #     print (clusters)

            plotgraph(clusters,"Kmeans ")
            return kmeans.labels_
         #     print ("Clusters using Kmeans: %d"%(len(set(kmeans.labels_))))

In [52]: predicted_labels = completefun(data)
```




```
In [53]: print (predicted_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 0 0 2 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2
 2 2 0 0 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 0 2
 2 0]
```

```
In [54]: print(labels)
```

```
['Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
```

```
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica']
```

```
In [55]: image = data[0]
```

```
In [56]: from matplotlib import pyplot as plt
         from sklearn.decomposition import TruncatedSVD
```

```
In [62]: svd = TruncatedSVD(n_components=2)
         scd_data = svd.fit_transform(data)
         tnsepoints = svd.fit_transform(data)
         # plotgraph(clusters,"Original  ")
```

```
In [63]: y = set(labels)
         clusters  = {}
         data = np.array(data)
         for i in range(len(labels)):
         #     predicted_labels.append(kmeans.labels_[i])
             if(labels[i] not in clusters):
                 clusters[labels[i]] = []
             clusters[labels[i]].append(i)
         for i in clusters.keys():
             clusters[i] = tnsepoints[clusters[i]]
```

```
In [64]: type(data[0][0])
```

```
Out[64]: numpy.float64
```

```
In [65]: clusters
```

```
Out[65]: {'Iris-setosa': array([[5.91220352, 2.30344211],
                 [5.57207573, 1.97383104],
                 [5.4464847 , 2.09653267],
                 [5.43601924, 1.87168085],
                 [5.87506555, 2.32934799],
                 [6.47699043, 2.32552598],
                 [5.51542859, 2.07156181],
                 [5.85042297, 2.14948016],
                 [5.15851287, 1.77642658],
                 [5.64458172, 1.99190598],
                 [6.2648267 , 2.42727947],
                 [5.75150446, 2.02142409],
                 [5.48017099, 1.97972378],
                 [4.95064168, 2.04947793],
                 [6.52528515, 2.9177484 ],
                 [6.78960484, 2.8254999 ],
                 [6.27173006, 2.60908548],
                 [5.92899145, 2.26896626],
                 [6.64757408, 2.39126214],
                 [6.09424306, 2.36154331],
                 [6.25352038, 2.12203279],
                 [6.07305216, 2.27257849],
                 [5.36933796, 2.4985751 ],
                 [6.04055493, 1.878367  ],
                 [5.90544974, 1.80875447],
                 [5.74982272, 1.86063438],
                 [5.93531393, 2.0096386 ],
                 [6.03863542, 2.26113533],
                 [5.94934149, 2.27753623],
                 [5.60042998, 1.88386305],
                 [5.63756795, 1.85795717],
                 [6.18446606, 2.19486085],
                 [6.24972051, 2.62254502],
                 [6.4785226 , 2.79919732],
                 [5.64458172, 1.99190598],
                 [5.62052002, 2.25317183],
                 [6.16135565, 2.48866436],
                 [5.64458172, 1.99190598],
                 [5.14517661, 1.90180543],
                 [5.92553978, 2.17806326],
                 [5.80255956, 2.31127304],
                 [4.97122949, 1.51448985],
                 [5.22113428, 2.01078338],
                 [6.00686864, 1.99517588],
                 [6.31629137, 2.04350797],
                 [5.51374686, 1.9107721 ],
                 [6.12877022, 2.32512928],
                 [5.42268299, 1.9970597 ],
```

```
       [6.1897099 , 2.39869637],
       [5.76112904, 2.16588106]]),
'Iris-versicolor': array([[ 9.12033960e+00, -7.00219921e-02],
       [ 8.58379652e+00, -1.34216664e-01],
       [ 9.12666208e+00, -3.29349657e-01],
       [ 7.27578441e+00, -4.58464245e-01],
       [ 8.55831307e+00, -3.94479344e-01],
       [ 7.87248767e+00, -4.83302548e-01],
       [ 8.66607667e+00, -2.84566378e-01],
       [ 6.45349296e+00,  2.41828100e-02],
       [ 8.63783285e+00, -2.42455583e-01],
       [ 7.16782218e+00, -2.89843601e-01],
       [ 6.47932460e+00, -3.06969745e-01],
       [ 7.97830954e+00, -1.73440471e-01],
       [ 7.56302580e+00, -2.66610207e-01],
       [ 8.33035185e+00, -4.90736782e-01],
       [ 7.37351386e+00,  1.80612202e-01],
       [ 8.70306507e+00,  2.40936792e-03],
       [ 7.90690441e+00, -4.71859382e-01],
       [ 7.65400146e+00, -1.22221396e-01],
       [ 8.05377454e+00, -7.36272611e-01],
       [ 7.34196793e+00, -1.81061634e-01],
       [ 8.41252158e+00, -5.93229301e-01],
       [ 7.91637942e+00, -1.45207918e-02],
       [ 8.44808823e+00, -8.27782086e-01],
       [ 8.25879715e+00, -4.76274069e-01],
       [ 8.33365396e+00, -8.69521519e-02],
       [ 8.58996943e+00, -8.06627036e-02],
       [ 8.86950574e+00, -4.16033961e-01],
       [ 9.02334060e+00, -5.80846389e-01],
       [ 8.16939279e+00, -4.12015974e-01],
       [ 7.23301526e+00,  2.20045780e-01],
       [ 7.17755719e+00, -1.93243831e-01],
       [ 7.10945416e+00, -8.78781122e-02],
       [ 7.58494714e+00, -4.93933360e-02],
       [ 8.41811361e+00, -9.80809017e-01],
       [ 7.75667080e+00, -5.29025574e-01],
       [ 8.37607491e+00, -1.74046941e-01],
       [ 8.87379829e+00, -2.44736100e-01],
       [ 8.08197922e+00, -5.13358976e-01],
       [ 7.66806816e+00, -1.19348195e-01],
       [ 7.35174208e+00, -3.49486294e-01],
       [ 7.57819336e+00, -5.44080972e-01],
       [ 8.31701559e+00, -3.65357932e-01],
       [ 7.59828340e+00, -1.74772186e-01],
       [ 6.49063093e+00, -1.72306959e-03],
       [ 7.60544675e+00, -3.53704996e-01],
       [ 7.77771213e+00, -1.27179129e-01],
```

```
       [ 7.75652123e+00, -2.16143949e-01],
       [ 8.18342035e+00, -1.44118344e-01],
       [ 6.50454806e+00,  3.14031756e-01],
       [ 7.66722730e+00, -1.99743050e-01]]),
'Iris-virginica': array([[ 9.4842643 , -1.51641735],
       [ 8.3182438 , -1.14140274],
       [ 9.8527954 , -1.24242625],
       [ 9.00957304, -1.20948284],
       [ 9.36756741, -1.3775108 ],
       [10.58758508, -1.5957399 ],
       [ 7.22476846, -1.01333762],
       [10.11994674, -1.419881  ],
       [ 9.2607551 , -1.45488611],
       [10.32556714, -1.16659243],
       [ 9.05074356, -0.70335204],
       [ 8.87157482, -1.11168392],
       [ 9.42218461, -1.04461604],
       [ 8.13264217, -1.24254976],
       [ 8.44016231, -1.25929299],
       [ 9.12862074, -0.97714242],
       [ 9.14647039, -1.0269378 ],
       [11.0346356 , -1.23661071],
       [10.69830769, -2.06673401],
       [ 8.1601164 , -1.14788818],
       [ 9.70946514, -1.11778643],
       [ 8.12014678, -1.03677605],
       [10.62127137, -1.71254878],
       [ 8.57440971, -0.82223167],
       [ 9.5636345 , -1.05151196],
       [10.00482117, -1.07232755],
       [ 8.48595665, -0.72543592],
       [ 8.53811261, -0.71593093],
       [ 9.09707481, -1.33881625],
       [ 9.79265744, -0.97057406],
       [10.07124246, -1.33848298],
       [10.99734807, -0.89782321],
       [ 9.11386274, -1.3732921 ],
       [ 8.66465493, -0.80609491],
       [ 8.67825118, -1.29221258],
       [10.43970228, -1.28165912],
       [ 9.30019482, -1.14389303],
       [ 9.10933242, -1.00103192],
       [ 8.41168071, -0.67362416],
       [ 9.48396516, -0.8906541 ],
       [ 9.48672553, -1.19302757],
       [ 9.36359574, -0.74693616],
       [ 8.3182438 , -1.14140274],
       [ 9.73697852, -1.28814928],
```

```
                [ 9.63078623, -1.18941534],
                [ 9.22669839, -0.92948121],
                [ 8.56655506, -1.03657534],
                [ 9.02610098, -0.88321986],
                [ 9.1056599 , -0.99622053],
                [ 8.49050919, -0.91487688]])}
```

In [66]:
```
# tnsepoints = svd.fit_transform(data)
plotgraph(clusters,"Original  ")
```

2 clusters are close to each other but are easily saperable but 1 cluster is fr from others

In [ ]:

In [ ]: