

```
In [1]: pip install pygad
```

```
Requirement already satisfied: pygad in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages (3.0.1)
Requirement already satisfied: cloudpickle in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages
(from pygad) (2.2.1)
Requirement already satisfied: matplotlib in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages (f
rom pygad) (3.7.1)
Requirement already satisfied: numpy in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages (from p
ygad) (1.23.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packa
ges (from matplotlib->pygad) (1.0.7)
Requirement already satisfied: cyclar>=0.10 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib->pygad) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-pack
ages (from matplotlib->pygad) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-pack
ages (from matplotlib->pygad) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packag
es (from matplotlib->pygad) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages
(from matplotlib->pygad) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packa
ges (from matplotlib->pygad) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-p
ackages (from matplotlib->pygad) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\dinesh reddy\appdata\local\programs\python\python310\lib\site-packages (fro
m python-dateutil>=2.7->matplotlib->pygad) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

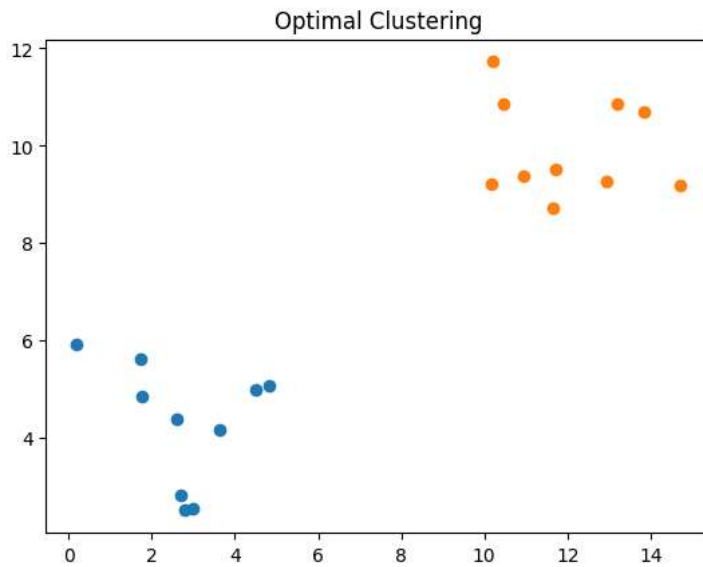
```
In [2]: import numpy
import matplotlib.pyplot
import pygad
```

```
In [3]: cluster1_num_samples = 10
cluster1_x1_start = 0
cluster1_x1_end = 5
cluster1_x2_start = 2
cluster1_x2_end = 6
cluster1_x1 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x1 = cluster1_x1 * (cluster1_x1_end - cluster1_x1_start) + cluster1_x1_start
cluster1_x2 = numpy.random.random(size=(cluster1_num_samples))
cluster1_x2 = cluster1_x2 * (cluster1_x2_end - cluster1_x2_start) + cluster1_x2_start
cluster2_num_samples = 10
cluster2_x1_start = 10
cluster2_x1_end = 15
cluster2_x2_start = 8
cluster2_x2_end = 12
cluster2_x1 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x1 = cluster2_x1 * (cluster2_x1_end - cluster2_x1_start) + cluster2_x1_start
cluster2_x2 = numpy.random.random(size=(cluster2_num_samples))
cluster2_x2 = cluster2_x2 * (cluster2_x2_end - cluster2_x2_start) + cluster2_x2_start
```

```
In [4]: c1 = numpy.array([cluster1_x1, cluster1_x2]).T
c2 = numpy.array([cluster2_x1, cluster2_x2]).T
data = numpy.concatenate((c1, c2), axis=0)
data
```

```
Out[4]: array([[ 1.73883127,  5.61569978],
 [ 0.18208438,  5.91522795],
 [ 3.63916083,  4.17051292],
 [ 2.59736505,  4.38399039],
 [ 2.69051094,  2.8136454 ],
 [ 4.51333221,  4.99342184],
 [ 1.75073854,  4.84784027],
 [ 4.8245605 ,  5.07879297],
 [ 2.79593071,  2.51085754],
 [ 2.97176003,  2.55388056],
 [12.93525074,  9.25890018],
 [10.2158543 , 11.72398548],
 [13.20371307, 10.86554269],
 [11.71705116,  9.52917503],
 [10.46144851, 10.85598446],
 [13.84899118, 10.69502032],
 [10.17466972,  9.22523038],
 [14.70525655,  9.18087037],
 [10.93446726,  9.37765178],
 [11.66574487,  8.71376996]])
```

```
In [5]: matplotlib.pyplot.scatter(cluster1_x1, cluster1_x2)
matplotlib.pyplot.scatter(cluster2_x1, cluster2_x2)
matplotlib.pyplot.title("Optimal Clustering")
matplotlib.pyplot.show()
```



```
In [6]: def euclidean_distance(X, Y):
return numpy.sqrt(numpy.sum(numpy.power(X - Y, 2), axis=1))
```

```
In [8]: def cluster_data(solution, solution_idx):
global num_cluster, data
feature_vector_length = data.shape[1]
cluster_centers = []
all_clusters_dists = []
clusters = []
clusters_sum_dist = []
for clust_idx in range(num_clusters):
    cluster_centers.append(solution[feature_vector_length*clust_idx:feature_vector_length*(clust_idx+1)])
    cluster_center_dists = euclidean_distance(data, cluster_centers[clust_idx])
    all_clusters_dists.append(numpy.array(cluster_center_dists))
cluster_centers = numpy.array(cluster_centers)
all_clusters_dists = numpy.array(all_clusters_dists)
cluster_indices = numpy.argmin(all_clusters_dists, axis=0)
for clust_idx in range(num_clusters):
    clusters.append(numpy.where(cluster_indices == clust_idx)[0])

    if len(clusters[clust_idx]) == 0:
        clusters_sum_dist.append(0)
    else:
        clusters_sum_dist.append(numpy.sum(all_clusters_dists[clust_idx, clusters[clust_idx]]))
clusters_sum_dist = numpy.array(clusters_sum_dist)
return cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist
```

```
In [9]: def fitness_func(ga_instance, solution, solution_idx):
_, _, _, _, clusters_sum_dist = cluster_data(solution, solution_idx)
fitness = 1.0 / (numpy.sum(clusters_sum_dist) + 0.00000001)
return fitness
```

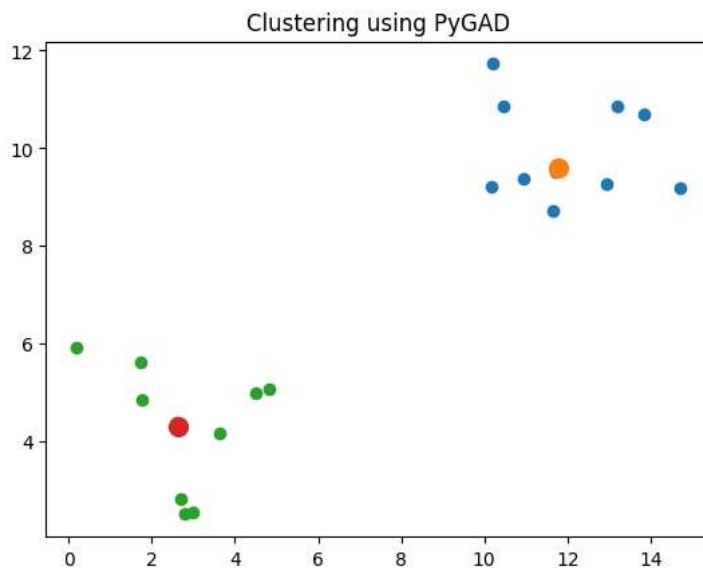
```
In [10]: num_clusters = 2
num_genes = num_clusters * data.shape[1]
ga_instance = pygad.GA(num_generations=100,
    sol_per_pop=10,
    num_parents_mating=5,
    init_range_low=-6,
    init_range_high=20,
    keep_parents=2,
    num_genes=num_genes,
    fitness_func=fitness_func,
    suppress_warnings=True)
ga_instance.run()
```

```
In [11]: best_solution, best_solution_fitness, best_solution_idx = ga_instance.best_solution()
print("Best solution is {bs}".format(bs=best_solution))
print("Fitness of the best solution is {bsf}".format(bsf=best_solution_fitness))
print("Best solution found after {gen} generations".format(gen=ga_instance.best_solution_generation))
```

```
Best solution is [11.76497192  9.58762794  2.64450235  4.31588802]
Fitness of the best solution is 0.0308444039284079746
Best solution found after 89 generations
```

```
In [13]: cluster_centers, all_clusters_dists, cluster_indices, clusters, clusters_sum_dist = cluster_data(best_solution, best_solution_idx)
```

```
In [14]: for cluster_idx in range(num_clusters):
    cluster_x = data[clusters[cluster_idx], 0]
    cluster_y = data[clusters[cluster_idx], 1]
    matplotlib.pyplot.scatter(cluster_x, cluster_y)
    matplotlib.pyplot.scatter(cluster_centers[cluster_idx, 0], cluster_centers[cluster_idx, 1], linewidths=5)
matplotlib.pyplot.title("Clustering using PyGAD")
matplotlib.pyplot.show()
```



```
In [ ]:
```