

```
In [44]: import pandas as pd
import math
import numpy as np
from array import array
import matplotlib.pyplot as plt
from statistics import mode
```

```
In [2]: data1=pd.read_csv("seeds.csv")
data1.head(10)
```

```
Out[2]:
```

	area	perimeter	compactness	lengthOfKernel	widthOfKernel	asymmetryCoefficient	lei
0	15.26	14.84	0.8710	5.763	3.312	2.221	
1	14.88	14.57	0.8811	5.554	3.333	1.018	
2	14.29	14.09	0.9050	5.291	3.337	2.699	
3	13.84	13.94	0.8955	5.324	3.379	2.259	
4	16.14	14.99	0.9034	5.658	3.562	1.355	
5	14.38	14.21	0.8951	5.386	3.312	2.462	
6	14.69	14.49	0.8799	5.563	3.259	3.586	
7	14.11	14.10	0.8911	5.420	3.302	2.700	
8	16.63	15.46	0.8747	6.053	3.465	2.040	
9	16.44	15.25	0.8880	5.884	3.505	1.969	

```
In [3]: data1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 8 columns):
area                210 non-null float64
perimeter           210 non-null float64
compactness         210 non-null float64
lengthOfKernel      210 non-null float64
widthOfKernel       210 non-null float64
asymmetryCoefficient 210 non-null float64
lengthOfKernelGroove 210 non-null float64
seedType            210 non-null int64
dtypes: float64(7), int64(1)
memory usage: 13.2 KB
```

```
In [4]: data1['seedType'].value_counts()
```

```
Out[4]: 3    70
2    70
1    70
Name: seedType, dtype: int64
```

```
In [5]: data1.shape
```

```
Out[5]: (210, 8)
```

```
In [6]: data1.describe()
```

```
Out[6]:
```

	area	perimeter	compactness	lengthOfKernel	widthOfKernel	asymmetryCoe
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.

```
In [7]: data=data1.iloc[:, :-1].values
data_seed=data1.iloc[:, -1].values
```

```
In [8]: def euclidean_distance(data):
        res=[]
        for i in range(len(data)):
            dist=[]
            for j in range (len(data)):
                temp=((data[i][0]-data[j][0])*2+(data[i][1]-data[j][1])*2+(da
ta[i][2]-data[j][2])*2)*0.5
                dist.append(temp)
            res.append(dist)
        return res
from sklearn.metrics.pairwise import euclidean_distances
```

[illegible]

```

In [10]: def seed_clustering(data, clusters):
    eucl_dist=euclidean_distances(data)
    #print(eucl_dist)
    eucl_dist = np.tril(eucl_dist)
    eucl_dist[eucl_dist == 0] = np.inf
    data_frame = pd.DataFrame(data=np.ones(data.shape[0])*np.inf)
    #if Linkage==1: #complete linkage
    d = {}
    for i in range(0, clusters):
        cluster_loop = np.unravel_index(eucl_dist.argmin(), eucl_dist.shape)

        if i == 0:
            data_frame.iloc[cluster_loop[0]] = 0
            data_frame.iloc[cluster_loop[1]] = 0
        else:
            try:
                a = int(data_frame.iloc[cluster_loop[0]])
            except:
                data_frame.iloc[cluster_loop[0]] = i
                a = i
            try:
                b = int(data_frame.iloc[cluster_loop[1]])
            except:
                data_frame.iloc[cluster_loop[1]] = i
                b = i
            data_frame[(data_frame[0]==a) | (data_frame[0]==b)] = i
        d[i] = cluster_loop
        for j in range(0, cluster_loop[0]):
            if np.isfinite(eucl_dist[cluster_loop[0]][j]) and np.isfinite(
eucl_dist[cluster_loop[1]][j]):
                eucl_dist[cluster_loop[1]][j] = max(eucl_dist[cluster_loo
p[0]][j], eucl_dist[cluster_loop[1]][j])
                eucl_dist[cluster_loop[0]] = np.inf
    return d, data_frame[0].as_matrix()

```

```

In [11]: forming_clusters5, rest_feature1 = seed_clustering(data, 206)

```

C:\Users\Administrator\Anaconda3\lib\site-packages\ipykernel_launcher.p
y:31: FutureWarning: Method .as_matrix will be removed in a future versi
on. Use .values instead.

In [12]: `forming_clusters5`

```
Out[12]: {0: (206, 172),
1: (198, 148),
2: (133, 122),
3: (28, 7),
4: (138, 137),
5: (49, 34),
6: (67, 33),
7: (209, 200),
8: (92, 91),
9: (14, 13),
10: (130, 123),
11: (163, 147),
12: (103, 96),
13: (127, 101),
14: (47, 44),
15: (68, 66),
16: (193, 177),
17: (182, 162),
18: (208, 192),
19: (131, 73),
20: (172, 168),
21: (169, 153),
22: (157, 152),
23: (55, 34),
24: (104, 92),
25: (202, 191),
26: (200, 143),
27: (158, 144),
28: (181, 166),
29: (166, 152),
30: (7, 2),
31: (75, 71),
32: (128, 90),
33: (174, 149),
34: (57, 1),
35: (58, 0),
36: (117, 73),
37: (176, 173),
38: (160, 148),
39: (45, 11),
40: (178, 173),
41: (21, 7),
42: (177, 175),
43: (161, 144),
44: (5, 2),
45: (106, 91),
46: (156, 153),
47: (122, 100),
48: (183, 150),
49: (168, 154),
50: (41, 40),
51: (48, 47),
52: (191, 172),
53: (201, 65),
54: (79, 37),
55: (125, 116),
56: (65, 42),
57: (173, 155),
58: (107, 76),
59: (155, 145),
```

```
60: (111, 96),
61: (119, 108),
62: (53, 20),
63: (148, 69),
64: (190, 149),
65: (115, 97),
66: (136, 107),
67: (118, 104),
68: (95, 74),
69: (44, 38),
70: (56, 5),
71: (204, 194),
72: (185, 154),
73: (194, 161),
74: (132, 43),
75: (50, 32),
76: (114, 77),
77: (52, 50),
78: (159, 150),
79: (109, 86),
80: (186, 150),
81: (91, 73),
82: (43, 10),
83: (147, 19),
84: (134, 43),
85: (187, 172),
86: (69, 26),
87: (189, 175),
88: (162, 150),
89: (33, 11),
90: (195, 181),
91: (205, 192),
92: (46, 17),
93: (72, 71),
94: (126, 125),
95: (34, 0),
96: (76, 70),
97: (9, 8),
98: (110, 99),
99: (105, 99),
100: (123, 91),
101: (98, 85),
102: (66, 54),
103: (179, 23),
104: (180, 176),
105: (112, 99),
106: (38, 6),
107: (135, 31),
108: (150, 143),
109: (59, 23),
110: (25, 22),
111: (184, 142),
112: (96, 87),
113: (35, 9),
114: (61, 60),
115: (139, 133),
116: (197, 63),
117: (170, 164),
118: (164, 145),
119: (24, 0),
```

120: (199, 65),
121: (99, 97),
122: (29, 13),
123: (42, 27),
124: (13, 7),
125: (40, 14),
126: (20, 7),
127: (121, 107),
128: (152, 147),
129: (17, 4),
130: (207, 203),
131: (171, 170),
132: (63, 19),
133: (196, 195),
134: (192, 156),
135: (36, 35),
136: (116, 102),
137: (100, 74),
138: (3, 2),
139: (64, 30),
140: (94, 78),
141: (22, 4),
142: (84, 73),
143: (167, 161),
144: (175, 174),
145: (11, 3),
146: (89, 77),
147: (124, 31),
148: (74, 71),
149: (101, 85),
150: (93, 81),
151: (165, 146),
152: (32, 6),
153: (80, 75),
154: (154, 144),
155: (120, 82),
156: (108, 84),
157: (142, 140),
158: (146, 61),
159: (54, 49),
160: (15, 12),
161: (137, 36),
162: (90, 83),
163: (97, 86),
164: (81, 78),
165: (26, 19),
166: (18, 11),
167: (153, 60),
168: (149, 145),
169: (188, 170),
170: (102, 96),
171: (129, 72),
172: (27, 26),
173: (86, 83),
174: (37, 36),
175: (31, 6),
176: (62, 26),
177: (87, 84),
178: (51, 43),
179: (82, 77),


```

180: (85, 73),
181: (144, 19),
182: (71, 70),
183: (113, 93),
184: (143, 140),
185: (12, 6),
186: (6, 2),
187: (141, 39),
188: (10, 6),
189: (140, 63),
190: (151, 145),
191: (88, 82),
192: (8, 4),
193: (145, 143),
194: (30, 11),
195: (19, 12),
196: (16, 15),
197: (1, 0),
198: (4, 1),
199: (60, 59),
200: (70, 37),
201: (39, 16),
202: (83, 73),
203: (203, 151),
204: (78, 70),
205: (7, 11)

```

In [13]: rest_feature1

```

Out[13]: array([[205., 205., 205., 205., 205., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 205., 205., 205., 205., 205., 205., 205.,
205., 199., 205., 205., 205., 205., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 205., 205., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 205., 199., 199., 199., 205., 205., 205., 205.,
205., 205., 205., 205., 205., 205., 205., 202., 205., 205., 205.,
191., 205., 205., 205., 205., 191., 202., 202., 202., 202., 202.,
191., 191., 202., 202., 202., 205., 205., 205., 202., 202., 202.,
202., 205., 202., 202., 202., 202., 202., 202., 205., 202., 202.,
202., 202., 202., 205., 191., 202., 202., 202., 202., 202., 191.,
205., 205., 202., 205., 202., 202., 202., 202., 205., 202., 202.,
205., 205., 205., 205., 205., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 199., 205., 205., 205., 205., 205., 205., 199.,
205., 205., 199., 205., 205., 205., 205., 205., 205., 205., 205.,
199., 205., 205., 205., 199., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 199., 205., 205., 205., 205., 205., 205., 205.,
205., 205., 205., 205., 205., 199., 205., 205., 205., 205., 205.,
205., 205., 205., 205., 205., 205., 205., 199., 205., 205., 199.,
205.]])

```

```
In [14]: #4 clusters
rest_feature1[rest_feature1==205]=1
rest_feature1[rest_feature1==202]=2
rest_feature1[rest_feature1==199]=3
rest_feature1[rest_feature1==191]=4
rest_feature1
```

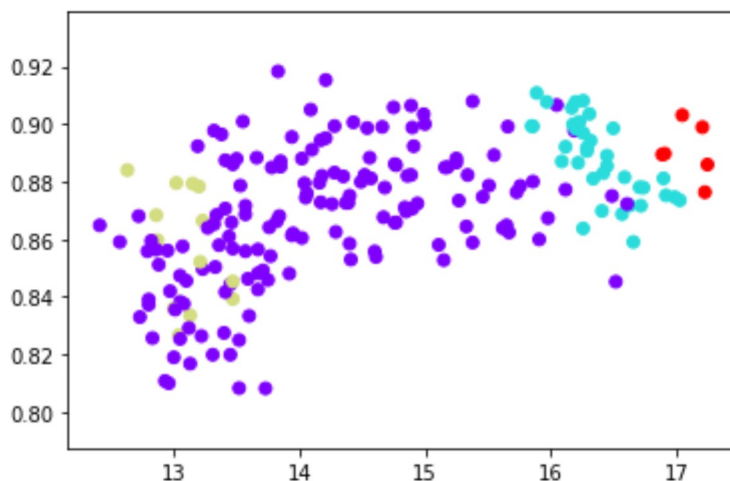
```
Out[14]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 1., 3., 1., 1., 1., 1., 1., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 1., 1., 1., 3., 3., 3., 1., 1., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 2., 1., 1., 1., 4., 1., 1., 1., 1., 4., 2.,
2.,
        2., 2., 2., 4., 4., 2., 2., 2., 1., 1., 1., 2., 2., 2., 2., 1.,
2.,
        2., 2., 2., 2., 2., 1., 2., 2., 2., 2., 2., 1., 4., 2., 2., 2.,
2.,
        2., 4., 1., 1., 2., 1., 2., 2., 2., 2., 1., 2., 2., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 3., 1., 1., 1., 1., 1.,
1.,
        3., 1., 1., 3., 1., 1., 1., 1., 1., 1., 1., 1., 3., 1., 1., 1.,
3.,
        1., 1., 1., 1., 1., 1., 1., 1., 1., 3., 1., 1., 1., 1., 1., 1.,
1.,
        1., 1., 1., 1., 1., 3., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1.,
        1., 3., 1., 1., 3., 1.]
```

```
In [36]: 100 - (final_res != rest_feature1).sum()/float(final_res.size)*100
```

```
Out[36]: 51.42857142857143
```

```
In [46]: plt.scatter(data[:,1], data[:,2], c=rest_feature1, cmap="rainbow")
```

```
Out[46]: <matplotlib.collections.PathCollection at 0x1cc0d140f08>
```



```
In [63]: forming_clusters10, rest_feature2 = seed_clustering(data, 208)
```

```
C:\Users\Administrator\Anaconda3\lib\site-packages\ipykernel_launcher.p  
y:31: FutureWarning: Method .as_matrix will be removed in a future versi  
on. Use .values instead.
```

In [64]: `forming_clusters10`

```
Out[64]: {0: (206, 172),
1: (198, 148),
2: (133, 122),
3: (28, 7),
4: (138, 137),
5: (49, 34),
6: (67, 33),
7: (209, 200),
8: (92, 91),
9: (14, 13),
10: (130, 123),
11: (163, 147),
12: (103, 96),
13: (127, 101),
14: (47, 44),
15: (68, 66),
16: (193, 177),
17: (182, 162),
18: (208, 192),
19: (131, 73),
20: (172, 168),
21: (169, 153),
22: (157, 152),
23: (55, 34),
24: (104, 92),
25: (202, 191),
26: (200, 143),
27: (158, 144),
28: (181, 166),
29: (166, 152),
30: (7, 2),
31: (75, 71),
32: (128, 90),
33: (174, 149),
34: (57, 1),
35: (58, 0),
36: (117, 73),
37: (176, 173),
38: (160, 148),
39: (45, 11),
40: (178, 173),
41: (21, 7),
42: (177, 175),
43: (161, 144),
44: (5, 2),
45: (106, 91),
46: (156, 153),
47: (122, 100),
48: (183, 150),
49: (168, 154),
50: (41, 40),
51: (48, 47),
52: (191, 172),
53: (201, 65),
54: (79, 37),
55: (125, 116),
56: (65, 42),
57: (173, 155),
58: (107, 76),
59: (155, 145),
```

```
60: (111, 96),
61: (119, 108),
62: (53, 20),
63: (148, 69),
64: (190, 149),
65: (115, 97),
66: (136, 107),
67: (118, 104),
68: (95, 74),
69: (44, 38),
70: (56, 5),
71: (204, 194),
72: (185, 154),
73: (194, 161),
74: (132, 43),
75: (50, 32),
76: (114, 77),
77: (52, 50),
78: (159, 150),
79: (109, 86),
80: (186, 150),
81: (91, 73),
82: (43, 10),
83: (147, 19),
84: (134, 43),
85: (187, 172),
86: (69, 26),
87: (189, 175),
88: (162, 150),
89: (33, 11),
90: (195, 181),
91: (205, 192),
92: (46, 17),
93: (72, 71),
94: (126, 125),
95: (34, 0),
96: (76, 70),
97: (9, 8),
98: (110, 99),
99: (105, 99),
100: (123, 91),
101: (98, 85),
102: (66, 54),
103: (179, 23),
104: (180, 176),
105: (112, 99),
106: (38, 6),
107: (135, 31),
108: (150, 143),
109: (59, 23),
110: (25, 22),
111: (184, 142),
112: (96, 87),
113: (35, 9),
114: (61, 60),
115: (139, 133),
116: (197, 63),
117: (170, 164),
118: (164, 145),
119: (24, 0),
```

120: (199, 65),
121: (99, 97),
122: (29, 13),
123: (42, 27),
124: (13, 7),
125: (40, 14),
126: (20, 7),
127: (121, 107),
128: (152, 147),
129: (17, 4),
130: (207, 203),
131: (171, 170),
132: (63, 19),
133: (196, 195),
134: (192, 156),
135: (36, 35),
136: (116, 102),
137: (100, 74),
138: (3, 2),
139: (64, 30),
140: (94, 78),
141: (22, 4),
142: (84, 73),
143: (167, 161),
144: (175, 174),
145: (11, 3),
146: (89, 77),
147: (124, 31),
148: (74, 71),
149: (101, 85),
150: (93, 81),
151: (165, 146),
152: (32, 6),
153: (80, 75),
154: (154, 144),
155: (120, 82),
156: (108, 84),
157: (142, 140),
158: (146, 61),
159: (54, 49),
160: (15, 12),
161: (137, 36),
162: (90, 83),
163: (97, 86),
164: (81, 78),
165: (26, 19),
166: (18, 11),
167: (153, 60),
168: (149, 145),
169: (188, 170),
170: (102, 96),
171: (129, 72),
172: (27, 26),
173: (86, 83),
174: (37, 36),
175: (31, 6),
176: (62, 26),
177: (87, 84),
178: (51, 43),
179: (82, 77),

207. (23 31)

rest_feature2

207.1)

