```
(*© 2012-Present Computational ClassNotes, Creative Commons License *)
(*https://creativecommons.org/licenses/by-nc-sa/3.0/us/  :
  Attribution-NonCommercial-ShareAlike *)
```

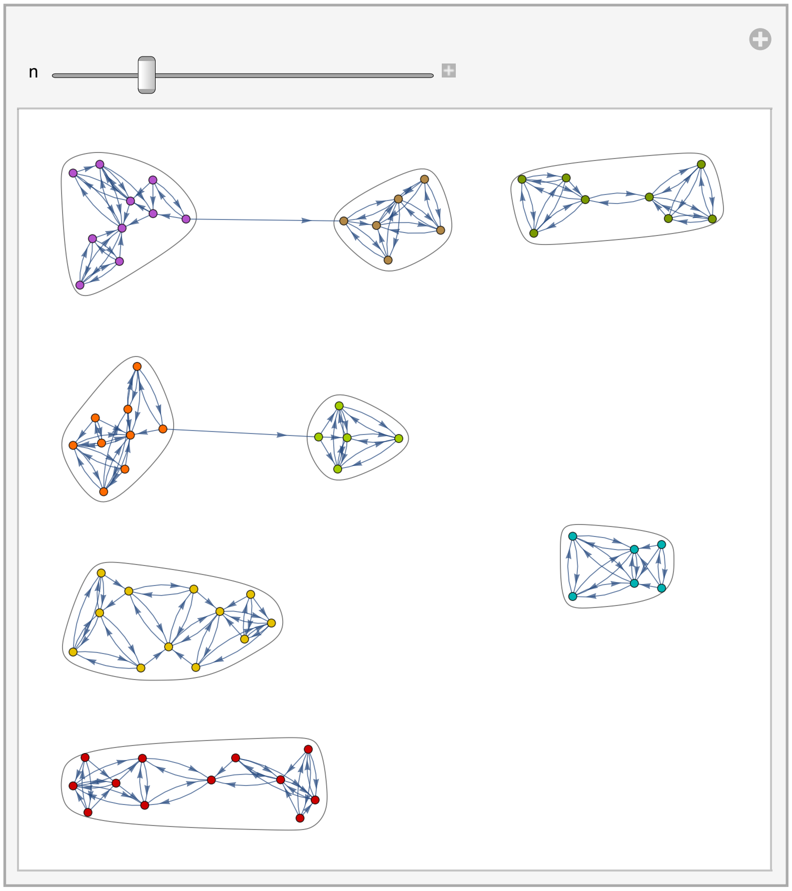
```
In[•]:= testdata = Import[
           FileNameJoin[{NotebookDirectory[], "aggregatefeaturesConorTest.csv"}], "CSV"];
        data = Import[FileNameJoin[{NotebookDirectory[],
             "aggregatefeaturesConorTrain.csv"}], "CSV"];
        Length@data
        Length@testdata

Out[•]= 68

Out[•]= 67
```

```
In[•]:= Dimensions@data
        Short@data

Out[•]= {68, 10}

Out[•]//Short= {{298.07, 0, 0.14803, 1.427, 6.1786, 1.2856, 5.3646, 25.02, 37.126, 2.9},
                 <<66>>, {119.68, 0, 0.8179, <<4>>, 6.059, 11.052, 14.4}}
```

Shape of Data

```
In[•]:= Manipulate[
         CommunityGraphPlot@
          NearestNeighborGraph[data, n, DistanceFunction → EuclideanDistance],

         {n, 1, 10, 1}, SaveDefinitions → True]
```

Out[•]=



Cluster Analysis

2 Cluster found!

```
In[•]:= clusters = FindClusters[data];
        Length@clusters

Out[•]= 2

In[•]:= Short@clusters[[1]]

Out[•]//Short= {{298.07, 0, 0.14803, 1.427, 6.1786, 1.2856, 5.3646, 25.02, 37.126, 2.9},
                 ≪58≫, {543.79, 0, 1.3584, ≪4≫, 27.318, 41.8, 14.4}}
```

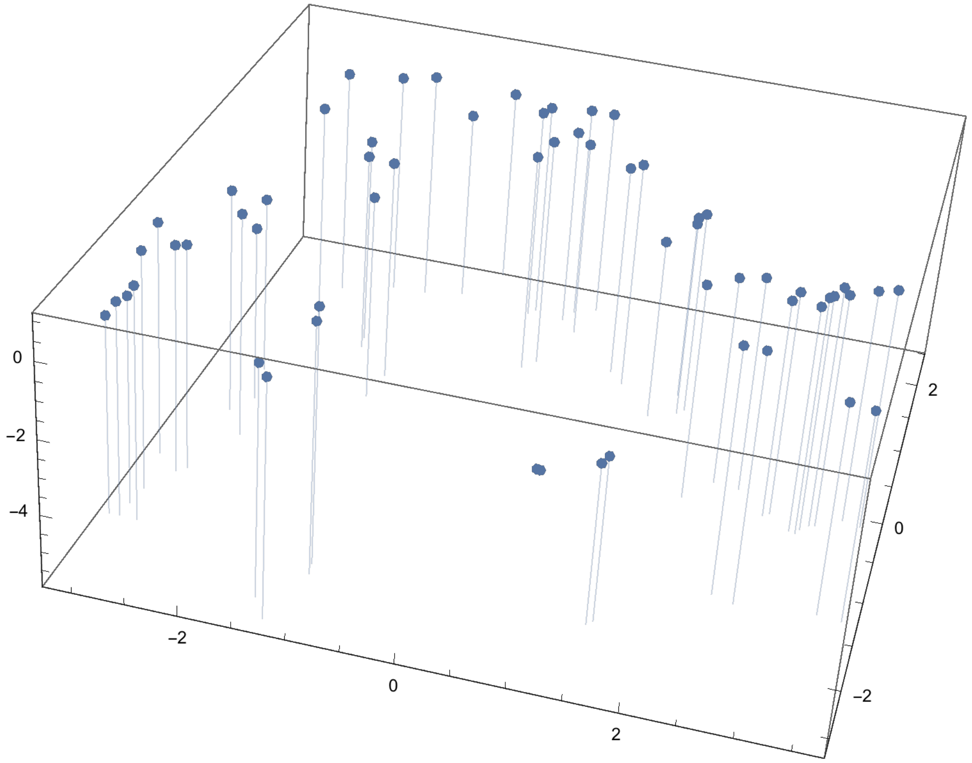
In[∘]:=
```
Short@clusters[[2]]
```

Out[∘]//Short=
```
{{75.422, 0, 0, 0.17126, 0.79574, 0.1713, 1.937, 5.2538, 9.5958, 2.9},
  {«1»}, «5», {119.68, 0, 0.8179, «4», 6.059, 11.052, 14.4}}
```

Cluster 1

In[∘]:=
```
reduced = DimensionReduce[clusters[[1]], 3];
ListPointPlot3D[reduced, ImageSize → 500, PlotRange → All, Filling → Bottom]
```

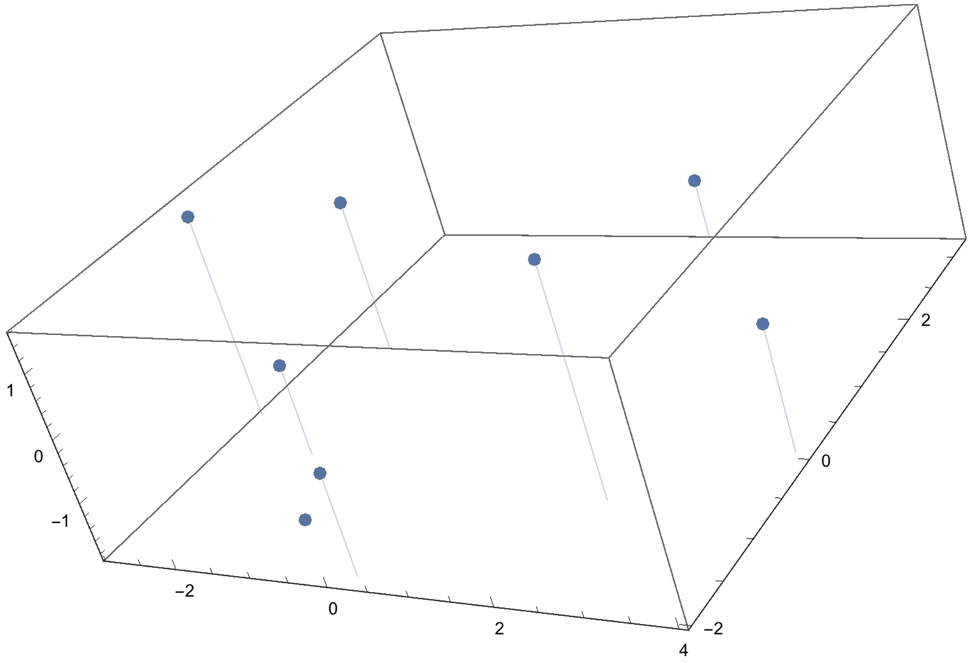
Out[∘]=



Cluster 2

In[•]:=
```
reduced = DimensionReduce[clusters[[2]], 3];
ListPointPlot3D[reduced, ImageSize → 500, PlotRange → All, Filling → Bottom]
```

Out[•]=

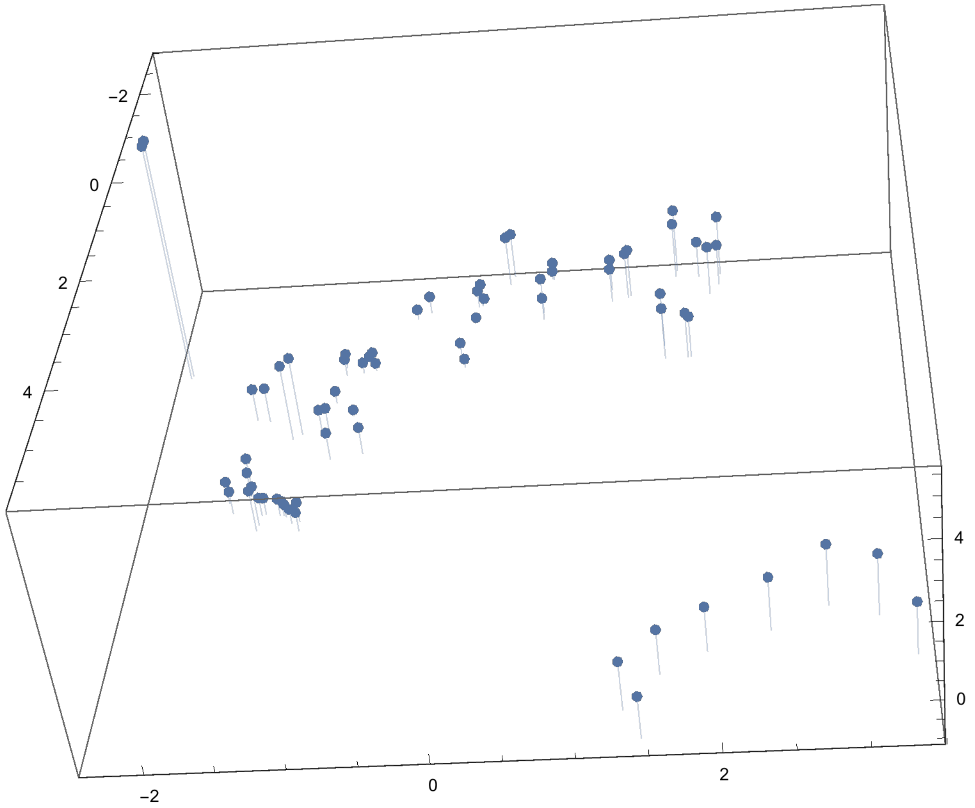


Both

Notice a possible outlier

```
In[•]:= reduced = DimensionReduce[data, 3];
        ListPointPlot3D[reduced, ImageSize → 500, PlotRange → All, Filling → Bottom]
```

Out[•]=



{x1,x2,x3...xn}⟶y

```
In[•]:=
        fxy = Flatten@Map[{Drop[#, 1] -> First@#} &, data];
        testfxy = Flatten@Map[{Drop[#, 1] -> First@#} &, testdata];
```

```
In[•]:= Short@fxy
```

```
Out[•]//Short= {{0, 0.14803, 1.427, 6.1786, 1.2856, 5.3646, 25.02, 37.126, 2.9} → 298.07,
                ≪66≫, {0, 0.8179, 2.1511, ≪3≫, 6.059, 11.052, 14.4} → ≪7≫}
```

```
In[•]:= Short@testfxy
```

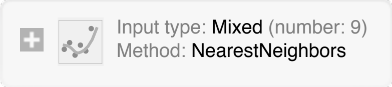
```
Out[•]//Short= {{0, 0.076604, 1.4922, 3.0296, 2.4288, 7.7585, 27.326, 36.57, 2.9} → 350.95,
                ≪65≫, {0.0000249, 0.79776, ≪5≫, 40.084, 14.4} → 529.7}
```

```
In[•]:= (*https://en.wikipedia.org/wiki/Coefficient_of_determination*)
        rSQUARED[y_, yhat_] := Module[{ybar, SStot, SSreg, SSres},

           ybar = Mean@y;
           SStot = Total@Map[(# - ybar) ^2 &, y];
           SSreg = Total@Map[(# - ybar) ^2 &, yhat];
           SSres = Total@Table[(y[[i]] - yhat[[i]]) ^2, {i, 1, Length@y}];

           1 - (SSres / SStot)

          ]
```

NearestNeighbors

```
In[•]:=
        p = Predict[fxy, Method → "NearestNeighbors", PerformanceGoal → "Quality"]
        rSQUARED[Table[testfxy[[i]][[2]], {i, 1, Length@testdata}],
         Table[p[testfxy[[i]][[1]]], {i, 1, Length@testdata}]]
```

Out[•]= PredictorFunction[



Input type: Mixed (number: 9)
Method: NearestNeighbors

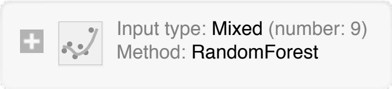
]

```
Out[•]= 0.898989
```

RandomForest

```
In[•]:= p = Predict[fxy, Method → "RandomForest", PerformanceGoal → "Quality"]
        rSQUARED[Table[testfxy[[i]][[2]], {i, 1, Length@testdata}],
         Table[p[testfxy[[i]][[1]]], {i, 1, Length@testdata}]]
```

Out[•]= PredictorFunction[



Input type: Mixed (number: 9)
Method: RandomForest

]

```
Out[•]= 0.923472
```

GaussianProcess

```
In[∘]:= p = Predict[fxy, Method → "GaussianProcess", PerformanceGoal → "Quality"]
        rSQUARED[Table[testfxy[[i]][[2]], {i, 1, Length@testdata}],
         Table[p[testfxy[[i]][[1]]], {i, 1, Length@testdata}]]

Out[∘]= PredictorFunction[ ⊞  Input type: Mixed (number: 9)
                               Method: GaussianProcess      ]

Out[∘]= 0.911278
```