SciPy

```
from scipy import constants
print(constants.pi)
```

```
3.141592653589793
```

```
from scipy.optimize import root
from math import cos
def eqn(x):
  return x + cos(x)
myroot = root(eqn, 0)
print(myroot.x)
```

```
[-0.73908513]
```

SciPy Graphs

```
import numpy as np
from scipy.sparse.csgraph import connected_components
from scipy.sparse import csr_matrix
arr = np.array([
  [0, 1, 2],
  [1, 0, 0],
  [2, 0, 0]
])
newarr = csr_matrix(arr)
print(connected_components(newarr))
```

```
(1, array([0, 0, 0], dtype=int32))
```

```
import numpy as np
from scipy.sparse.csgraph import floyd_warshall
from scipy.sparse import csr_matrix
arr = np.array([
  [0, 1, 2],
  [1, 0, 0],
  [2, 0, 0]
])
newarr = csr_matrix(arr)
print(floyd_warshall(newarr, return_predecessors=True))
```

```
(array([[0., 1., 2.],
       [1., 0., 3.],
       [2., 3., 0.]]), array([[-9999,     0,     0],
       [    1, -9999,     0],
       [    2,     0, -9999]], dtype=int32))
```

-

Sklearn

```
import numpy as np
from sklearn.preprocessing import MinMaxScaler
demoData = np.random.randint(10, 100, (10 ,2))
demoData
```

```
scalar_model = MinMaxScaler()
scalar_model.fit_transform(demoData)
```

```
array([[0.58571429, 0.60240964],
       [0.4       , 0.68674699],
       [0.44285714, 1.        ],
       [1.        , 0.60240964],
       [0.64285714, 0.09638554],
       [0.31428571, 0.26506024],
       [0.41428571, 0.        ],
       [0.        , 0.46987952],
       [0.24285714, 0.38554217],
       [0.92857143, 0.01204819]])
```

```
import numpy as np
from sklearn.preprocessing import MinMaxScaler
```

+ Code + Text

```
demoData = np.random.randint(1, 500, (20 ,4))
demoData
```

```
array([[154, 401, 332, 335],
       [362, 251, 397, 462],
       [293,  78, 372, 139],
       [ 64,  87, 425,  60],
       [178, 495,  10,  30],
       [ 59, 169, 295, 475],
       [274, 453,  65, 344],
       [115, 419, 332,   1],
       [403, 452, 480, 362],
       [237, 253, 105, 418],
       [319, 475, 191, 448],
       [ 43, 265, 432, 362],
       [ 73, 212, 494,   3],
       [108, 295,  87,  26],
       [ 61, 358, 403, 276],
       [ 87, 364, 381, 347],
       [283, 484, 355, 428],
       [ 89,  83, 109, 224],
       [ 31, 279, 470, 183],
       [427, 304, 138, 182]])
```

```
scalar_model = MinMaxScaler()
feature_data = scalar_model.fit_transform(demoData)
feature_data
```

```
array([[0.31060606, 0.77458034, 0.66528926, 0.70464135],
```

```
[0.83585859, 0.41486811, 0.79958678, 0.97257384],
[0.66161616, 0.        , 0.74793388, 0.29113924],
[0.08333333, 0.02158273, 0.85743802, 0.12447257],
[0.37121212, 1.        , 0.        , 0.06118143],
[0.07070707, 0.21822542, 0.58884298, 1.        ],
[0.61363636, 0.89928058, 0.11363636, 0.72362869],
[0.21212121, 0.8177458 , 0.66528926, 0.        ],
[0.93939394, 0.89688249, 0.97107438, 0.76160338],
[0.52020202, 0.41966427, 0.19628099, 0.87974684],
[0.72727273, 0.95203837, 0.37396694, 0.94303797],
[0.03030303, 0.44844125, 0.87190083, 0.76160338],
[0.10606061, 0.32134293, 1.        , 0.00421941],
[0.19444444, 0.52038369, 0.15909091, 0.05274262],
[0.07575758, 0.67146283, 0.81198347, 0.58016878],
[0.14141414, 0.68585132, 0.76652893, 0.72995781],
[0.63636364, 0.9736211 , 0.71280992, 0.90084388],
[0.14646465, 0.01199041, 0.20454545, 0.47046414],
[0.        , 0.48201439, 0.95041322, 0.38396624],
[1.        , 0.54196643, 0.26446281, 0.38185654]])
```

```
import pandas as pd
df = pd.DataFrame(data=feature_data, columns=['k1', 'k2', 'k3', 'labels'])
df
```

	k1	k2	k3	labels
0	0.310606	0.774580	0.665289	0.704641
1	0.835859	0.414868	0.799587	0.972574
2	0.661616	0.000000	0.747934	0.291139
3	0.083333	0.021583	0.857438	0.124473
4	0.371212	1.000000	0.000000	0.061181
5	0.070707	0.218225	0.588843	1.000000
6	0.613636	0.899281	0.113636	0.723629
7	0.212121	0.817746	0.665289	0.000000
8	0.939394	0.896882	0.971074	0.761603
9	0.520202	0.419664	0.196281	0.879747
10	0.727273	0.952038	0.373967	0.943038
11	0.030303	0.448441	0.871901	0.761603
12	0.106061	0.321343	1.000000	0.004219
13	0.194444	0.520384	0.159091	0.052743