Data Scaling

1. Normalization

Formula : x_scaled = (x - min) / (max - min)

Example 1

In [1]:
```
# Generate data
import pandas as pd
df = pd.DataFrame({'a':[100,8,50,88,4],'b':[0.001,0.02,0.009,0.07,0.1]})
```

In [2]:
```
df
```

Out[2]:

	a	b
0	100	0.001
1	8	0.020
2	50	0.009
3	88	0.070
4	4	0.100

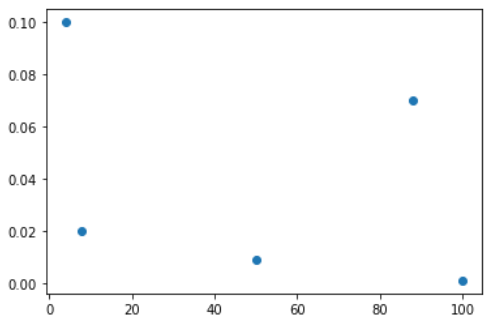
In [3]:
```
df.describe()
```

Out[3]:

	a	b
count	5.000000	5.000000
mean	50.000000	0.040000
std	44.226689	0.042959
min	4.000000	0.001000
25%	8.000000	0.009000
50%	50.000000	0.020000
75%	88.000000	0.070000
max	100.000000	0.100000

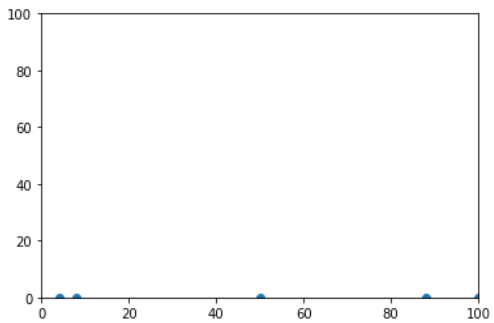
In [4]:
```
# Plot data
import matplotlib.pyplot as plt
plt.scatter(df['a'],df['b'])
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x7f99b1486be0>
```



In [5]:
```
plt.xlim(0,100)
plt.ylim(0,100)
plt.scatter(df['a'],df['b'])
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x7f99b15a0b80>
```



In [6]:
```
# Define min max scaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [7]:
```
# transform data
scaled = scaler.fit_transform(df)
```

In [8]:
```
scaled
```

Out[8]:
```
array([[1.        , 0.        ],
       [0.04166667, 0.19191919],
       [0.47916667, 0.08080808],
       [0.875     , 0.6969697 ],
       [0.        , 1.        ]])
```

In [9]:
```
df_scaled = pd.DataFrame(scaled)
```

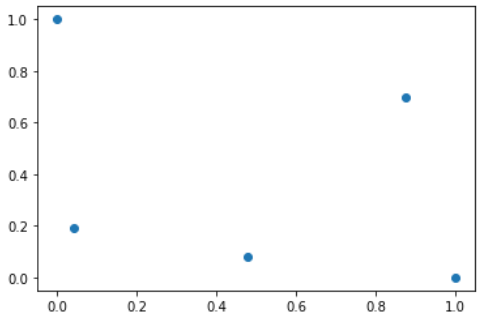
In [10]:
```
df_scaled
```

Out[10]:

	0	1
0	1.000000	0.000000
1	0.041667	0.191919
2	0.479167	0.080808
3	0.875000	0.696970
4	0.000000	1.000000

In [11]:
```
plt.scatter(df_scaled[0],df_scaled[1])
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x7f99b26b9a30>
```



In [12]:
```
df_scaled.describe()
```

Out[12]:

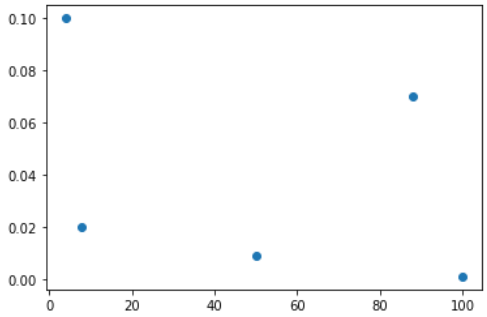
	0	1
count	5.000000	5.000000
mean	0.479167	0.393939
std	0.460695	0.433932
min	0.000000	0.000000
25%	0.041667	0.080808
50%	0.479167	0.191919
75%	0.875000	0.696970
max	1.000000	1.000000

Example 2

In [13]:
```
# Generate data
from numpy import asarray
data = asarray([[100, 0.001], [8, 0.02], [50, 0.009], [88, 0.07], [4, 0.1]])
```

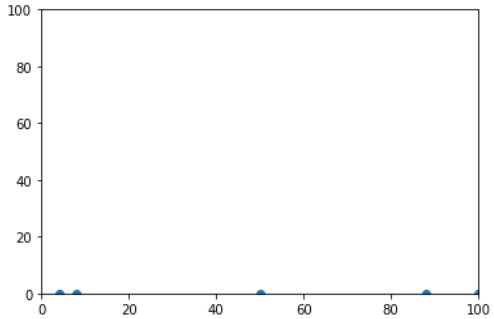
In [14]:
```
# Plot data
import matplotlib.pyplot as plt
plt.scatter(data[:,0],data[:,1])
```

Out[14]: <matplotlib.collections.PathCollection at 0x7f99b27837c0>



In [15]:
```
plt.xlim(0,100)
plt.ylim(0,100)
plt.scatter(data[:,0],data[:,1])
```

Out[15]: <matplotlib.collections.PathCollection at 0x7f99b285dfa0>

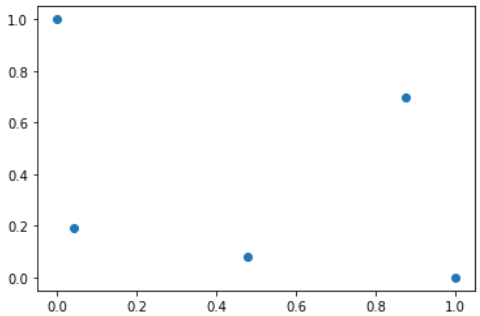


In [16]:
```
# Define min max scaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [17]:
```
# Transform data
scaled = scaler.fit_transform(data)
```

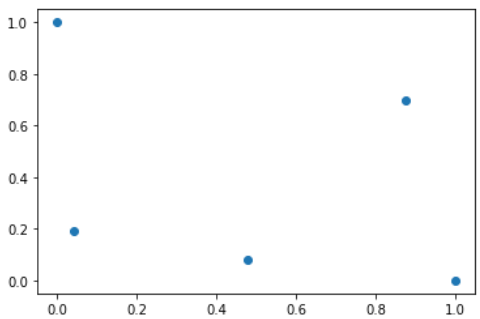
In [18]:
```
plt.scatter(scaled[:,0],scaled[:,1])
```

Out[18]: <matplotlib.collections.PathCollection at 0x7f99b2944f10>



In [19]:
```
a_scaled = [i[0] for i in scaled]
b_scaled = [i[1] for i in scaled]
plt.scatter(a_scaled,b_scaled)
```

Out[19]: <matplotlib.collections.PathCollection at 0x7f99b2a35a30>



2. Standardization

Formula : x_scaled = (x - mean) / SD

In [20]:
```
# Generate data
import pandas as pd
df = pd.DataFrame({'a':[100,8,50,88,4],'b':[0.001,0.02,0.009,0.07,0.1]})
```

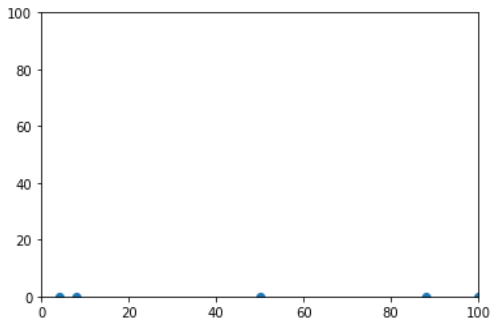
In [21]:
```
df
```

Out[21]:

	a	b
0	100	0.001
1	8	0.020
2	50	0.009
3	88	0.070
4	4	0.100

In [22]:
```
# Plot data
import matplotlib.pyplot as plt
plt.xlim(0,100)
plt.ylim(0,100)
plt.scatter(df['a'],df['b'])
```

Out[22]:
```
<matplotlib.collections.PathCollection at 0x7f99b2b30280>
```



In [23]:
```
df.describe()
```

Out[23]:

	a	b
count	5.000000	5.000000
mean	50.000000	0.040000
std	44.226689	0.042959
min	4.000000	0.001000
25%	8.000000	0.009000
50%	50.000000	0.020000
75%	88.000000	0.070000
max	100.000000	0.100000

In [24]:
```
# define standard scaler
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [25]:
```
# transform data
scaled = scaler.fit_transform(df)
```

In [26]:
```
scaled
```

Out[26]:
```
array([[ 1.26398112, -1.01499193],
       [-1.06174414, -0.52050868],
       [ 0.        , -0.80678846],
       [ 0.96062565,  0.78076302],
       [-1.16286263,  1.56152604]])
```

In [27]:
```
df_scaled = pd.DataFrame(scaled)
```

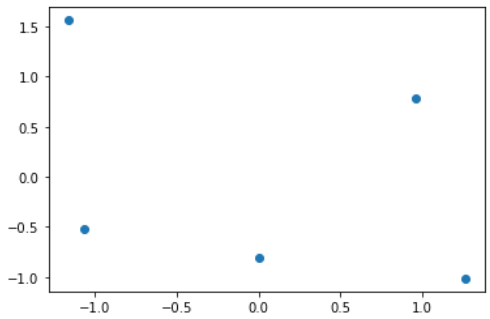
In [28]:
```
df_scaled
```

Out[28]:

	0	1
0	1.263981	-1.014992
1	-1.061744	-0.520509
2	0.000000	-0.806788
3	0.960626	0.780763
4	-1.162863	1.561526

In [29]:
```
plt.scatter(df_scaled[0],df_scaled[1])
```

Out[29]:
```
<matplotlib.collections.PathCollection at 0x7f99b2c15730>
```



In [30]:
```
df_scaled.describe()
```

Out[30]:

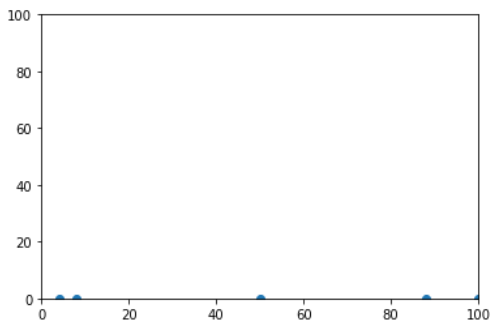
	0	1
count	5.000000e+00	5.000000
mean	4.440892e-17	0.000000
std	1.118034e+00	1.118034
min	-1.162863e+00	-1.014992
25%	-1.061744e+00	-0.806788
50%	0.000000e+00	-0.520509
75%	9.606256e-01	0.780763
max	1.263981e+00	1.561526

3. Using "sklearn" library

In [31]:
```
# Generate data
import pandas as pd
df = pd.DataFrame({'a':[100,8,50,88,4],'b':[0.001,0.02,0.009,0.07,0.1]})
```

In [32]:
```
# Plot data
import matplotlib.pyplot as plt
plt.xlim(0,100)
plt.ylim(0,100)
plt.scatter(df['a'],df['b'])
```

Out[32]:
```
<matplotlib.collections.PathCollection at 0x7f99b2c94880>
```



In [33]:
```
df.describe()
```

Out[33]:

	a	b
count	5.000000	5.000000
mean	50.000000	0.040000
std	44.226689	0.042959
min	4.000000	0.001000
25%	8.000000	0.009000
50%	50.000000	0.020000
75%	88.000000	0.070000
max	100.000000	0.100000

In [34]:
```
from sklearn import preprocessing
# transform data
scaled = preprocessing.scale(df)
```

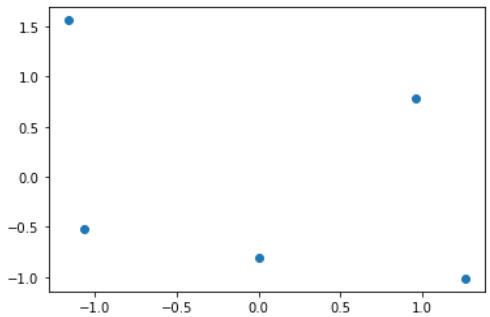
In [35]:
```
scaled
```

Out[35]:
```
array([[ 1.26398112, -1.01499193],
       [-1.06174414, -0.52050868],
       [ 0.        , -0.80678846],
       [ 0.96062565,  0.78076302],
       [-1.16286263,  1.56152604]])
```

In [36]:
```
df_scaled = pd.DataFrame(scaled)
```

In [37]:
```
plt.scatter(df_scaled[0],df_scaled[1])
```

Out[37]:
```
<matplotlib.collections.PathCollection at 0x7f99b2dd0940>
```



In [38]:
```
df_scaled.describe()
```

Out[38]:

	0	1
count	5.000000e+00	5.000000
mean	4.440892e-17	0.000000
std	1.118034e+00	1.118034
min	-1.162863e+00	-1.014992
25%	-1.061744e+00	-0.806788
50%	0.000000e+00	-0.520509
75%	9.606256e-01	0.780763
max	1.263981e+00	1.561526

In [ ]:
```
```