In [1]:

```
#Rescale Data
```

In [83]:

```
# importing libraries
import pandas as pd
import scipy
import numpy
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
```

In [3]:

```
# data set link
dataset = "customers.csv"
df = pd.read_csv(dataset)
df.head()
```

Out[3]:

	CustomerID	Age	Annual Income ($)	Spending Score (1-100)	Work Experience	Family Size
0	1	19	15000	39	1	4
1	2	21	35000	81	3	3
2	3	20	86000	6	1	1
3	4	23	59000	77	0	2
4	5	31	38000	40	2	6

In [4]:

```
# data parameters
names = ['Age', 'Annual Income ($)', 'Spending Score (1-100)' ,'Work Experience', 'Family
```

In [5]:

```
# preparating of dataframe using the data and defined columns list
dataframe = pd.read_csv(dataset, names=names, header=0) # added header=0 to skip fi
array = dataframe.values
```

In [10]:

```
array
```

Out[10]:

```
array([[    19,  15000,     39,      1,      4],
       [    21,  35000,     81,      3,      3],
       [    20,  86000,      6,      1,      1],
       ...,
       [    87,  90961,     14,      9,      2],
       [    77, 182109,      4,      7,      2],
       [    90, 110610,     52,      5,      2]], dtype=int64)
```

In [31]:

```
# separate array into input and output components
# .iloc[rows, columns]
# .iloc[1:4, 0:5]
# it means rows 1-3 and columns 0 to 4
X = dataframe.iloc[:, 0:5].values
Y = dataframe.iloc[:, 4].values
```

In [32]:

```
X
```

Out[32]:

```
array([[    19,  15000,     39,      1,      4],
       [    21,  35000,     81,      3,      3],
       [    20,  86000,      6,      1,      1],
       ...,
       [    87,  90961,     14,      9,      2],
       [    77, 182109,      4,      7,      2],
       [    90, 110610,     52,      5,      2]], dtype=int64)
```

In [14]:

```
Y
```

Out[14]:

```
array([4, 3, 1, ..., 2, 2, 2], dtype=int64)
```

In [37]:

```
# initialising the MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
```

In [38]:

```
# learning the statistical parameters for each of the data and transforming
rescaledX = scaler.fit_transform(X)
```

In [39]:

```
# summarize transformed data
numpy.set_printoptions(precision=2) # precision is number of decimals
print(rescaledX[0:5,:])          # 5 is the number of rows to be printed
```

```
[[0.19 0.08 0.39 0.06 0.38]
 [0.21 0.18 0.81 0.18 0.25]
 [0.2  0.45 0.06 0.06 0.  ]
 [0.23 0.31 0.77 0.   0.12]
 [0.31 0.2  0.4  0.12 0.62]]
```

In [40]:

```
print(rescaledX[0:4,:])
```

```
[[0.19 0.08 0.39 0.06 0.38]
 [0.21 0.18 0.81 0.18 0.25]
 [0.2  0.45 0.06 0.06 0.  ]
 [0.23 0.31 0.77 0.   0.12]]
```

In [41]:

```
# Binarize Data
```

In [42]:

```
# import libraries
from sklearn.preprocessing import Binarizer
import pandas
import numpy
```

In [43]:

```
# data set link
dataset = "customers.csv"
```

In [44]:

```
# data parameters
names = ['Age', 'Annual Income ($)', 'Spending Score (1-100)' ,'Work Experience', 'Family
```

In [45]:

```
# preparating of dataframe using the data and defined columns list
dataframe = pd.read_csv(dataset, names=names, header=0) # added header=0 to skip fi
array = dataframe.values
```

In [51]:

```
# separate array into input and output components
X = array[:, 0:5]
Y = array[:, 4]
```

In [52]:

```
# threshold - values below or equal to this are replaced by 0, above it by 1.
binarizer = Binarizer(threshold = 0.0).fit(X)
binaryX = binarizer.transform(X)
```

In [79]:

```
# summarize transformed data
print(binaryX[0:5,])
```

```
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 0 1]
 [1 1 1 1 1]]
```

In [74]:

```
# Standard Scalar
```

In [75]:

```
# importing libraries
from sklearn.preprocessing import StandardScaler
import pandas
import numpy
```

In [59]:

```
# data set link
dataset = "customers.csv"
```

In [60]:

```
# data parameters
names = ['Age', 'Annual Income ($)', 'Spending Score (1-100)' ,'Work Experience', 'Family
```

In [61]:

```
# preparating of dataframe using the data and defined columns list
dataframe = pd.read_csv(dataset, names=names, header=0) # added header=0 to skip fi
array = dataframe.values
```

In [80]:

```
# separate array into input and output components
X = array[:, 0:5]
Y = array[:, 4]
```

In [84]:

```
# initialising the StandardScaler
scaler = StandardScaler().fit(X)
```

In [85]:

```
rescaledX = scaler.transform(X)
```

In [86]:

```
# summarize transformed data
numpy.set_printoptions(precision = 3)
print(rescaledX[0:5,:])
```

```
[[-1.054 -2.094 -0.428 -0.791  0.117]
 [-0.984 -1.656  1.076 -0.281 -0.39 ]
 [-1.019 -0.541 -1.61  -0.791 -1.405]
 [-0.913 -1.131  0.932 -1.046 -0.898]
 [-0.632 -1.591 -0.393 -0.536  1.133]]
```

In [ ]:

```

```