In [105]:

```
import numpy as np
import pandas as pd
```

In [106]:

```
df = pd.read_csv("test.csv")
df.head()
```

Out[106]:

	id	battery_power	blue	clock_speed	dual_sim	int_memory	mobile_wt	n_cores	pc	px_h
0	1	1043	1	1.8	1	5	193	3	16	
1	2	841	1	0.5	1	61	191	5	12	
2	3	1807	1	2.8	0	27	186	3	4	
3	4	1546	0	0.5	1	25	96	8	20	
4	5	1434	0	1.4	0	49	108	6	18	

In [107]:

```
#Set beneficial and Non beneficial criteria to all
beneficial = df[['battery_power','blue','clock_speed','dual_sim','int_memory','n_cores','pc
                 ,'ram','sc_h','three_g','touch_screen','wifi']]
print(beneficial.head())
```

```
   battery_power  blue  clock_speed  dual_sim  int_memory  n_cores  pc  \
0           1043     1          1.8         1           5        3  16
1            841     1          0.5         1          61        5  12
2           1807     1          2.8         0          27        3   4
3           1546     0          0.5         1          25        8  20
4           1434     0          1.4         0          49        6  18

   px_height  px_width   ram  sc_h  three_g  touch_screen  wifi
0        226      1412  3476    12        0             1     0
1        746       857  3895     6        1             0     0
2       1270      1366  2396    17        0             1     1
3        295      1752  3893    10        1             1     0
4        749       810  1773    15        1             0     1
```

In [108]:

```
Non_beneficial = df[['mobile_wt']]
print(Non_beneficial.head())
```

```
   mobile_wt
0        193
1        191
2        186
3         96
4        108
```

In [109]:

```
for i in beneficial:
    column = beneficial[i]
    max=column.max()
#     print(max)
    beneficial[i] = (beneficial[i] / max)

beneficial
```

```
c:\users\abhay\appdata\local\programs\python\python37\lib\site-packages\ipyk
ernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  """
```

Out[109]:

	battery_power	blue	clock_speed	dual_sim	int_memory	n_cores	pc	px_height	px_w
0	0.521761	1.0	0.600000	1.0	0.078125	0.375	0.80	0.118511	0.706
1	0.420710	1.0	0.166667	1.0	0.953125	0.625	0.60	0.391190	0.428
2	0.903952	1.0	0.933333	0.0	0.421875	0.375	0.20	0.665967	0.683
3	0.773387	0.0	0.166667	1.0	0.390625	1.000	1.00	0.154693	0.876
4	0.717359	0.0	0.466667	0.0	0.765625	0.750	0.90	0.392764	0.405
...	...	...	...	...	...	...	...	...	
995	0.850425	1.0	0.633333	0.0	0.843750	0.875	0.85	0.337703	0.456
996	0.304652	0.0	0.600000	1.0	0.203125	0.500	0.10	0.604090	0.816
997	0.592796	0.0	0.466667	0.0	0.125000	0.125	0.60	0.250131	0.412
998	0.766883	1.0	0.166667	1.0	0.781250	0.250	0.60	0.019927	0.416
999	0.635318	1.0	0.166667	0.0	0.546875	0.750	0.95	0.239643	0.304

1000 rows × 14 columns

In [110]:

```
for i in Non_beneficial:
    column = Non_beneficial[i]
    min=column.min()
#     print(max)
    Non_beneficial[i] = (min / Non_beneficial[i])

Non_beneficial
```

```
c:\users\abhay\appdata\local\programs\python\python37\lib\site-packages\ipyk
ernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  """
```

Out[110]:

	mobile_wt
0	0.414508
1	0.418848
2	0.430108
3	0.833333
4	0.740741
...	...
995	0.470588
996	0.430108
997	1.000000
998	0.467836
999	0.571429

1000 rows × 1 columns

In [111]:

```
frames = [beneficial, Non_beneficial]
df2 = pd.concat([beneficial, Non_beneficial], axis=1)
df2
```

Out[111]:

	battery_power	blue	clock_speed	dual_sim	int_memory	n_cores	pc	px_height	px_w
0	0.521761	1.0	0.600000	1.0	0.078125	0.375	0.80	0.118511	0.706
1	0.420710	1.0	0.166667	1.0	0.953125	0.625	0.60	0.391190	0.428
2	0.903952	1.0	0.933333	0.0	0.421875	0.375	0.20	0.665967	0.683
3	0.773387	0.0	0.166667	1.0	0.390625	1.000	1.00	0.154693	0.876
4	0.717359	0.0	0.466667	0.0	0.765625	0.750	0.90	0.392764	0.405
...	...	...	...	...	...	...	...	...	
995	0.850425	1.0	0.633333	0.0	0.843750	0.875	0.85	0.337703	0.456
996	0.304652	0.0	0.600000	1.0	0.203125	0.500	0.10	0.604090	0.816
997	0.592796	0.0	0.466667	0.0	0.125000	0.125	0.60	0.250131	0.412
998	0.766883	1.0	0.166667	1.0	0.781250	0.250	0.60	0.019927	0.416
999	0.635318	1.0	0.166667	0.0	0.546875	0.750	0.95	0.239643	0.304

1000 rows × 15 columns

In [112]:

```
# Set weight for each column
# weight=1/(len(df2. columns))
# for i in df2:
#     df2[i] = (df2[i] * weight)
# df2

#
weights = np.random.dirichlet(np.ones(15),size=1)
print(weights)
df2 = df2*weights
df2
```

```
[[0.03835213 0.01823431 0.00551114 0.00710582 0.07301131 0.07516191
  0.09981749 0.18038119 0.07558393 0.12013282 0.01222556 0.018569
  0.12436881 0.01101494 0.14052964]]
```

Out[112]:

	battery_power	blue	clock_speed	dual_sim	int_memory	n_cores	pc	px_height	px_width
0	0.020011	0.018234	0.003307	0.007106	0.005704	0.028186	0.079854	0.021377	0.053416
1	0.016135	0.018234	0.000919	0.007106	0.069589	0.046976	0.059890	0.070563	0.032420
2	0.034668	0.018234	0.005144	0.000000	0.030802	0.028186	0.019963	0.120128	0.051675
3	0.029661	0.000000	0.000919	0.007106	0.028520	0.075162	0.099817	0.027904	0.066278
4	0.027512	0.000000	0.002572	0.000000	0.055899	0.056371	0.089836	0.070847	0.030642
...	...	...	...	...	...	...	...	...	...
995	0.032616	0.018234	0.003490	0.000000	0.061603	0.065767	0.084845	0.060915	0.034539

In [113]:

```
df2["Score"] = df2.sum(axis=1)
df2
```

Out[113]:

	battery_power	blue	clock_speed	dual_sim	int_memory	n_cores	pc	px_heig
0	0.020011	0.018234	0.003307	0.007106	0.005704	0.028186	0.079854	0.0213
1	0.016135	0.018234	0.000919	0.007106	0.069589	0.046976	0.059890	0.0705
2	0.034668	0.018234	0.005144	0.000000	0.030802	0.028186	0.019963	0.1201
3	0.029661	0.000000	0.000919	0.007106	0.028520	0.075162	0.099817	0.0279
4	0.027512	0.000000	0.002572	0.000000	0.055899	0.056371	0.089836	0.0708
...	...	...	...	...	...	...	...	
995	0.032616	0.018234	0.003490	0.000000	0.061603	0.065767	0.084845	0.0609
996	0.011684	0.000000	0.003307	0.007106	0.014830	0.037581	0.009982	0.1089
997	0.022735	0.000000	0.002572	0.000000	0.009126	0.009395	0.059890	0.0451
998	0.029412	0.018234	0.000919	0.007106	0.057040	0.018790	0.059890	0.0035
999	0.024366	0.018234	0.000919	0.000000	0.039928	0.056371	0.094827	0.0432

1000 rows × 16 columns

In [115]:

```
df2.sort_values(by=['Score'], ascending=False)
```

Out[115]:

	battery_power	blue	clock_speed	dual_sim	int_memory	n_cores	pc	px_heig
716	0.032635	0.018234	0.002939	0.007106	0.060462	0.056371	0.069872	0.1211
168	0.022639	0.018234	0.005511	0.000000	0.049054	0.009395	0.079854	0.1472
865	0.033057	0.018234	0.001102	0.007106	0.065026	0.046976	0.094827	0.1298
143	0.028126	0.000000	0.004409	0.007106	0.065026	0.056371	0.089836	0.1448
563	0.036817	0.000000	0.001102	0.007106	0.071871	0.075162	0.099817	0.0729
...	...	...	...	...	...	...	...	
169	0.016346	0.018234	0.003490	0.000000	0.003422	0.018790	0.019963	0.0355
318	0.012586	0.000000	0.004960	0.007106	0.054758	0.018790	0.000000	0.0286
697	0.034246	0.018234	0.004593	0.000000	0.020534	0.037581	0.004991	0.0109
603	0.025594	0.000000	0.002204	0.007106	0.039928	0.018790	0.004991	0.0149
282	0.015080	0.000000	0.003307	0.007106	0.031942	0.009395	0.034936	0.0002

1000 rows × 16 columns

In [ ]: