

```
In [4]: import pandas as pd
import numpy as np
```

```
In [5]: temp = pd.read_csv("dataset_Facebook-1.csv", sep=';')
print(temp.index)
temp
```

RangeIndex(start=0, stop=500, step=1)

Out[5]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Cor
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	
4	139441	Photo	2	12	2	3	0.0	7244	13594	671	
...	
495	85093	Photo	3	1	7	2	0.0	4684	7536	733	
496	81370	Photo	2	1	5	8	0.0	3480	6229	537	
497	81370	Photo	1	1	5	2	0.0	3778	7216	625	
498	81370	Photo	3	1	4	11	0.0	4156	7564	626	
499	81370	Photo	2	1	4	4	NaN	4188	7292	564	

500 rows × 19 columns



```
In [3]: print(temp.columns)
```

```
Index(['Page total likes', 'Type', 'Category', 'Post Month', 'Post Weekday',
      'Post Hour', 'Paid', 'Lifetime Post Total Reach',
      'Lifetime Post Total Impressions', 'Lifetime Engaged Users',
      'Lifetime Post Consumers', 'Lifetime Post Consumptions',
      'Lifetime Post Impressions by people who have liked your Page',
      'Lifetime Post reach by people who like your Page',
      'Lifetime People who have liked your Page and engaged with your post',
      'comment', 'like', 'share', 'Total Interactions'],
      dtype='object')
```

In [4]: `print(temp.shape)`

(500, 19)

In [5]: `print(temp.describe())`

	Page total likes	Category	Post Month	Post Weekday	Post Hour \
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	123194.176000	1.880000	7.038000	4.150000	7.840000
std	16272.813214	0.852675	3.307936	2.030701	4.368589
min	81370.000000	1.000000	1.000000	1.000000	1.000000
25%	112676.000000	1.000000	4.000000	2.000000	3.000000
50%	129600.000000	2.000000	7.000000	4.000000	9.000000
75%	136393.000000	3.000000	10.000000	6.000000	11.000000
max	139441.000000	3.000000	12.000000	7.000000	23.000000

	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions
\			
count	499.000000	500.000000	5.000000e+02
mean	0.278557	13903.36000	2.958595e+04
std	0.448739	22740.78789	7.680325e+04
min	0.000000	238.00000	5.700000e+02
25%	0.000000	3315.00000	5.694750e+03
50%	0.000000	5281.00000	9.051000e+03
75%	1.000000	13168.00000	2.208550e+04
max	1.000000	139441.00000	1.110000e+05

```
In [6]: #create subset
selective_df = temp[{'Category','like','share'}]
print(selective_df)
print(selective_df.shape)

print(selective_df['Category'].values)
#print(selective_df['like'].values)
#print(np.unique(selective_df['like'].values).shape)
```

	share	Category	like
0	17.0	2	79.0
1	29.0	2	130.0
2	14.0	3	66.0
3	147.0	2	1572.0
4	49.0	2	325.0
..
495	26.0	3	53.0
496	22.0	2	53.0
497	18.0	1	93.0
498	38.0	3	91.0
499	28.0	2	91.0

[500 rows x 3 columns]

(500, 3)

```
[2 2 3 2 2 2 3 3 2 3 2 2 2 2 2 2 3 1 3 3 2 1 1 3 2 2 2 3 2 1 2 2 3 3 1 2 3
 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1
 1 1 1 1 2 1 2 3 1 3 3 1 1 3 1 1 1 2 3 3 2 3 1 2 2 1 1 2 3 1 1 1 3 2 3 2 1
 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 3 2 3 1 1 3 3
 2 1 1 2 2 1 2 2 2 3 2 2 3 2 3 2 1 2 1 3 1 2 3 1 2 1 2 2 3 1 2 2 2 1 3 1 2
 1 2 2 3 2 3 3 2 2 1 2 1 2 1 3 2 1 2 3 3 1 1 2 3 2 3 3 2 3 3 1 2 2 1 3 2 3
 2 2 1 2 3 1 2 2 3 3 2 1 2 2 3 2 1 2 3 1 2 1 2 1 2 2 3 2 1 1 3 2 3 2 1 1 2
 3 2 1 2 2 3 3 1 2 1 1 2 1 1 3 2 2 3 1 1 1 2 2 2 1 1 2 1 1 1 1 1 2 1 2 1 1
 3 3 2 1 2 3 1 3 1 3 2 3 3 3 1 3 3 3 1 3 3 2 3 2 1 3 3 3 3 3 2 3 3 3 3 3 3
 3 2 2 3 3 3 1 3 3 2 3 1 3 3 3 3 3 2 3 1 3 2 1 3 2 2 3 1 3 2 3 1 3 3 3 1 2
 1 2 1 1 1 3 3 3 1 3 1 3 1 3 1 3 1 1 1 1 3 2 1 3 3 1 3 1 3 3 3 1 1 1 3 3 3
 3 1 2 2 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 3 3
 2 1 1 1 1 1 1 1 1 3 3 3 1 1 3 2 3 1 3 3 1 3 2 2 1 3 1 3 1 3 1 3 1 1 3 3 2
 1 3 2 3 1 3 3 3 3 3 3 1 3 3 3 2 1 3 2]
```

```
In [7]: #reshaping
pivot_table = pd.pivot_table(selective_df, index= ['Category', 'like'])
print(pivot_table)
print(pivot_table.shape)
```

		share
Category	like	
1	0.0	0.000000
	1.0	2.000000
	2.0	0.000000
	3.0	0.666667
	4.0	1.000000
...		...
3	1155.0	102.000000
	1372.0	47.000000
	1546.0	181.000000
	1639.0	122.000000
	1998.0	128.000000

[368 rows x 1 columns]
(368, 1)

```
In [8]: pivot_table.shape
```

Out[8]: (368, 1)

```
In [9]: pivot_table.reset_index(inplace=True)
pivot_table
```

Out[9]:

	Category	like	share
0	1	0.0	0.000000
1	1	1.0	2.000000
2	1	2.0	0.000000
3	1	3.0	0.666667
4	1	4.0	1.000000
...
363	3	1155.0	102.000000
364	3	1372.0	47.000000
365	3	1546.0	181.000000
366	3	1639.0	122.000000
367	3	1998.0	128.000000

368 rows × 3 columns

```
In [10]: pivot_table.melt(id_vars=['like', 'share'])
```

```
Out[10]:
```

	like	share	variable	value
0	0.0	0.000000	Category	1
1	1.0	2.000000	Category	1
2	2.0	0.000000	Category	1
3	3.0	0.666667	Category	1
4	4.0	1.000000	Category	1
...
363	1155.0	102.000000	Category	3
364	1372.0	47.000000	Category	3
365	1546.0	181.000000	Category	3
366	1639.0	122.000000	Category	3
367	1998.0	128.000000	Category	3

368 rows × 4 columns

```
In [11]: data1 = {'Name': ['Jai', 'Hari', 'Gaurav', 'Anuj'],
                  'Age': [27, 24, 22, 32],
                  'Address': ['Nagpur', 'Kanpur', 'Allahabad', 'Kannada'],
                  'Qualification': ['Msc', 'MA', 'ME', 'Phd']}
data2 = {'Name': ['Jaya', 'Harish', 'Anuja', 'Tanuja'],
         'Age': [17, 14, 12, 42],
         'Address': ['Nagpur', 'Kanpur', 'Allahabad', 'Kannada'],
         'Qualification': ['Btech', 'BA', 'BSc', 'BArch']}
df = pd.DataFrame(data1, index=[0,1,2,3])
df1 = pd.DataFrame(data2, index=[4,5,6,7])
print(df, "\n\n", df1)
```

	Name	Age	Address	Qualification
0	Jai	27	Nagpur	Msc
1	Hari	24	Kanpur	MA
2	Gaurav	22	Allahabad	ME
3	Anuj	32	Kannada	Phd

	Name	Age	Address	Qualification
4	Jaya	17	Nagpur	Btech
5	Harish	14	Kanpur	BA
6	Anuja	12	Allahabad	BSc
7	Tanuja	42	Kannada	BArch

```
In [12]: frames = [df, df1]
result = pd.concat(frames)
print(result)
```

	Name	Age	Address	Qualification
0	Jai	27	Nagpur	Msc
1	Hari	24	Kanpur	MA
2	Gaurav	22	Allahabad	ME
3	Anuj	32	Kannada	Phd
4	Jaya	17	Nagpur	Btech
5	Harish	14	Kanpur	BA
6	Anuja	12	Allahabad	BSc
7	Tanuja	42	Kannada	BArch

```
In [13]: df_new = pd.DataFrame(data1, index=[0,1,2,3])
df1 = pd.DataFrame(data2, index=[2,3,4,5])
#print(df, "\n\n", df1)

result2 = pd.concat([df_new, df1], axis=1, join='inner') #by index
print(result2)
result2 = pd.concat([df_new, df1], axis=1, join='outer') #by index
print(result2)
```

	Name	Age	Address	Qualification		Name	Age	Address	Qualification
2	Gaurav	22	Allahabad	ME		Jaya	17	Nagpur	Btech
3	Anuj	32	Kannada	Phd		Harish	14	Kanpur	BA
	Name	Age	Address	Qualification		Name	Age	Address	\
0	Jai	27.0	Nagpur	Msc		NaN	NaN	NaN	
1	Hari	24.0	Kanpur	MA		NaN	NaN	NaN	
2	Gaurav	22.0	Allahabad	ME		Jaya	17.0	Nagpur	
3	Anuj	32.0	Kannada	Phd		Harish	14.0	Kanpur	
4	NaN	NaN	NaN	NaN		Anuja	12.0	Allahabad	
5	NaN	NaN	NaN	NaN		Tanuja	42.0	Kannada	

	Qualification
0	NaN
1	NaN
2	Btech
3	BA
4	BSc
5	BArch

```
In [14]: #Defining the dictionary
data1 = {'Key':['k0','k1','k2','k3'],
        'Name': ['Jai', 'Hari', 'Gaurav', 'Anuj'],
        'Age':[27, 24, 22, 32]}
data2 = {'Key':['k0','k1','k2','k3'],
        'Address':['Nagpur', 'Kanpur', 'Allahabad', 'Kannada'],
        'Qualification':['Btech', 'BA', 'BSc', 'BArch']}

#convert the dictionary into dataframe
df = pd.DataFrame(data1)
df1 = pd.DataFrame(data2)

print(df, '\n\n', df1)

res = pd.merge(df, df1, on = 'Key')
res
```

	Key	Name	Age
0	k0	Jai	27
1	k1	Hari	24
2	k2	Gaurav	22
3	k3	Anuj	32

	Key	Address	Qualification
0	k0	Nagpur	Btech
1	k1	Kanpur	BA
2	k2	Allahabad	BSc
3	k3	Kannada	BArch

```
Out[14]:
```

	Key	Name	Age	Address	Qualification
0	k0	Jai	27	Nagpur	Btech
1	k1	Hari	24	Kanpur	BA
2	k2	Gaurav	22	Allahabad	BSc
3	k3	Anuj	32	Kannada	BArch

```
In [15]: cars = ['Ford', 'BMW', 'Volvo']
cars.sort()
print(cars)
#List.sort(reverse=True/False, key=myFunc)

['BMW', 'Ford', 'Volvo']
```

```
In [16]: df = pd.DataFrame({'Weight':[45, 88, 56, 15, 71],  
                           'Name':['Sam', 'Andrea', 'Alex', 'Robin', 'Kia'],  
                           'Age':[14, 25, 55, 8, 21]})  
  
print(df)
```

	Weight	Name	Age
0	45	Sam	14
1	88	Andrea	25
2	56	Alex	55
3	15	Robin	8
4	71	Kia	21

```
In [17]: # return the transpose  
result = df.transpose()  
  
# Print the result  
print(result)
```

	0	1	2	3	4
Weight	45	88	56	15	71
Name	Sam	Andrea	Alex	Robin	Kia
Age	14	25	55	8	21

```
In [ ]:
```