

```
In [1]: # import pandas library
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
In [2]: # Reading csv file
data = pd.read_csv("dataset_Facebook.csv")
# creating a dataframe of it
data = pd.DataFrame(data)
data.head()
```

Out[2]:

	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	

```
In [3]: # Datatype of each column
data.dtypes
```

```
Out[3]: Page total likes                                int64
4
Type                                                    objec
t
Category                                                int64
4
Post Month                                              int64
4
Post Weekday                                             int64
4
Post Hour                                               int64
4
Paid                                                    float64
4
Lifetime Post Total Reach                              int64
4
Lifetime Post Total Impressions                        int64
4
Lifetime Engaged Users                                int64
4
Lifetime Post Consumers                                int64
4
Lifetime Post Consumptions                             int64
4
Lifetime Post Impressions by people who have liked your Page int64
4
Lifetime Post reach by people who like your Page       int64
4
Lifetime People who have liked your Page and engaged with your post int64
4
comment                                                 int64
4
like                                                    float64
4
share                                                  float64
4
Total Interactions                                     int64
4
dtype: object
```

In [4]: *# Information about each column data*
 data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 19 columns):
 #   Column                                     Non
-Null Count  Dtype
---  -
0   Page total likes                          500
non-null    int64
1   Type                                       500
non-null    object
2   Category                                  500
non-null    int64
3   Post Month                               500
non-null    int64
4   Post Weekday                             500
non-null    int64
5   Post Hour                                500
non-null    int64
6   Paid                                      499
non-null    float64
7   Lifetime Post Total Reach                 500
non-null    int64
8   Lifetime Post Total Impressions           500
non-null    int64
9   Lifetime Engaged Users                   500
non-null    int64
10  Lifetime Post Consumers                   500
non-null    int64
11  Lifetime Post Consumptions                500
non-null    int64
12  Lifetime Post Impressions by people who have liked your Page  500
non-null    int64
13  Lifetime Post reach by people who like your Page                500
non-null    int64
14  Lifetime People who have liked your Page and engaged with your post  500
non-null    int64
15  comment                                   500
non-null    int64
16  like                                      499
non-null    float64
17  share                                    496
non-null    float64
18  Total Interactions                       500
non-null    int64
dtypes: float64(3), int64(15), object(1)
memory usage: 74.3+ KB
```

```
In [5]: # Count ,min,max ,etc of each column
data.describe()
```

Out[5]:

	Page total likes	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach
count	500.000000	500.000000	500.000000	500.000000	500.000000	499.000000	500.000000
mean	123194.176000	1.880000	7.038000	4.150000	7.840000	0.278557	13903.36000
std	16272.813214	0.852675	3.307936	2.030701	4.368589	0.448739	22740.78789
min	81370.000000	1.000000	1.000000	1.000000	1.000000	0.000000	238.00000
25%	112676.000000	1.000000	4.000000	2.000000	3.000000	0.000000	3315.00000
50%	129600.000000	2.000000	7.000000	4.000000	9.000000	0.000000	5281.00000
75%	136393.000000	3.000000	10.000000	6.000000	11.000000	1.000000	13168.00000
max	139441.000000	3.000000	12.000000	7.000000	23.000000	1.000000	180480.00000

```
In [6]: #Finding null values in each column
data.isna().sum()
```

```
Out[6]: Page total likes      0
Type      0
Category  0
Post Month      0
Post Weekday    0
Post Hour      0
Paid          1
Lifetime Post Total Reach    0
Lifetime Post Total Impressions 0
Lifetime Engaged Users      0
Lifetime Post Consumers      0
Lifetime Post Consumptions   0
Lifetime Post Impressions by people who have liked your Page 0
Lifetime Post reach by people who like your Page      0
Lifetime People who have liked your Page and engaged with your post comment 0
like      1
share     4
Total Interactions      0
dtype: int64
```

In [7]: *# Replacing null values of Paid with it's mean*

```
paid_mean = data['Paid'].mean()
print(paid_mean)
data['Paid'].fillna(paid_mean, inplace=True)
data.isna().sum()
```

0.2785571142284569

```
Out[7]: Page total likes                                0
Type                                                    0
Category                                                0
Post Month                                              0
Post Weekday                                            0
Post Hour                                              0
Paid                                                    0
Lifetime Post Total Reach                             0
Lifetime Post Total Impressions                       0
Lifetime Engaged Users                                0
Lifetime Post Consumers                                0
Lifetime Post Consumptions                            0
Lifetime Post Impressions by people who have liked your Page 0
Lifetime Post reach by people who like your Page      0
Lifetime People who have liked your Page and engaged with your post 0
comment                                                0
like                                                    1
share                                                  4
Total Interactions                                    0
dtype: int64
```

```
In [8]: # Replacing null values of Like with it's mean
like_mean = data['like'].mean()
print(like_mean)
data['like'].fillna(like_mean, inplace=True)
data.isna().sum()
```

```
177.94589178356713
```

```
Out[8]: Page total likes                                0
Type                                                    0
Category                                                0
Post Month                                              0
Post Weekday                                            0
Post Hour                                              0
Paid                                                    0
Lifetime Post Total Reach                             0
Lifetime Post Total Impressions                       0
Lifetime Engaged Users                                0
Lifetime Post Consumers                                0
Lifetime Post Consumptions                            0
Lifetime Post Impressions by people who have liked your Page 0
Lifetime Post reach by people who like your Page       0
Lifetime People who have liked your Page and engaged with your post 0
comment                                                 0
like                                                    0
share                                                   4
Total Interactions                                     0
dtype: int64
```

```
In [9]: # Replacing null values of Paid with it's mean
share_mean = data['share'].mean()
print(share_mean)
data['share'].fillna(share_mean, inplace=True)
data.isna().sum()
```

27.266129032258064

```
Out[9]: Page total likes      0
Type      0
Category  0
Post Month      0
Post Weekday    0
Post Hour      0
Paid          0
Lifetime Post Total Reach    0
Lifetime Post Total Impressions 0
Lifetime Engaged Users      0
Lifetime Post Consumers      0
Lifetime Post Consumptions   0
Lifetime Post Impressions by people who have liked your Page 0
Lifetime Post reach by people who like your Page      0
Lifetime People who have liked your Page and engaged with your post 0
comment      0
like         0
share        0
Total Interactions      0
dtype: int64
```

```
In [10]: #Finding number of zeros in each column
for i in data:
    count = (data[i] == 0).sum()
    print('Zeros in column',i,'->', count)
```

```
Zeros in column Page total likes -> 0
Zeros in column Type -> 0
Zeros in column Category -> 0
Zeros in column Post Month -> 0
Zeros in column Post Weekday -> 0
Zeros in column Post Hour -> 0
Zeros in column Paid -> 360
Zeros in column Lifetime Post Total Reach -> 0
Zeros in column Lifetime Post Total Impressions -> 0
Zeros in column Lifetime Engaged Users -> 0
Zeros in column Lifetime Post Consumers -> 0
Zeros in column Lifetime Post Consumptions -> 0
Zeros in column Lifetime Post Impressions by people who have liked your Page
-> 0
Zeros in column Lifetime Post reach by people who like your Page -> 0
Zeros in column Lifetime People who have liked your Page and engaged with you
r post -> 0
Zeros in column comment -> 106
Zeros in column like -> 5
Zeros in column share -> 13
Zeros in column Total Interactions -> 6
```

```
In [11]: # splitting data using columns
data_frame1=data[['Page total likes','Type','Category','Post Month','Post Weekday','Post Hour','comment','like','share']]
data_frame1.head()
```

```
Out[11]:
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	comment	like	share
0	139441	Photo	2	12	4	3	4	79.0	17.0
1	139441	Status	2	12	3	10	5	130.0	29.0
2	139441	Photo	3	12	3	3	0	66.0	14.0
3	139441	Photo	2	12	2	10	58	1572.0	147.0
4	139441	Photo	2	12	2	3	19	325.0	49.0

```
In [12]: #splitting data using loc function
data_frame2=data.loc[:,['Page total likes','Paid']]
data_frame2.head()
```

```
Out[12]:
```

	Page total likes	Paid
0	139441	0.0
1	139441	0.0
2	139441	0.0
3	139441	1.0
4	139441	0.0

```
In [13]: # splitting data using iloc function using indexes of columns
data_frame3=data.iloc[:,[0,15,16,17]]
data_frame3.head()
```

```
Out[13]:
```

	Page total likes	comment	like	share
0	139441	4	79.0	17.0
1	139441	5	130.0	29.0
2	139441	0	66.0	14.0
3	139441	58	1572.0	147.0
4	139441	19	325.0	49.0


```
In [14]: # splitting data using train test split
x = data.drop(['Paid'], axis=1)
y = data.Paid
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
x_train.head(),x_test.head(),y_train.head(),y_test.head())
```

```
Out[14]: (    Page total likes    Type  Category  Post Month  Post Weekday  Post H
our \
107          136736  Status          2          10          1
4          336          119198  Photo          3          5          6
14          71          137893  Video          1          11          5
3          474          91009  Photo          1          2          1
12          6          139441  Photo          3          12          1
3

      Lifetime Post Total Reach  Lifetime Post Total Impressions \
107                        9504                        19556
336                        2772                        4642
71                        100768                       220447
474                        21928                        39641
6                         11692                        19479
```

```
In [15]: # Merging data_frame1 and data_frame2
data_frame3=pd.merge(data_frame1, data_frame2)
data_frame3.head()
```

```
Out[15]:
```

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	comment	like	share	Paid
0	139441	Photo	2	12	4	3	4	79.0	17.0	0.0
1	139441	Photo	2	12	4	3	4	79.0	17.0	0.0
2	139441	Photo	2	12	4	3	4	79.0	17.0	0.0
3	139441	Photo	2	12	4	3	4	79.0	17.0	1.0
4	139441	Photo	2	12	4	3	4	79.0	17.0	0.0

```
In [16]: # Find rows having Paid = 1
paid_data=data[data['Paid']==1.0]
paid_data.head()
```

Out[16]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Co
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	
14	138414	Photo	2	12	4	5	1.0	22784	39941	887	
17	138414	Photo	1	12	2	12	1.0	53264	111785	1706	

```
In [17]: #Finding rows having Likes > 750
likes_750=data[data['like']>750]
likes_750.head()
```

Out[17]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	C
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	
101	137020	Photo	2	10	4	3	0.0	68896	104952	2624	
105	137020	Photo	1	10	2	4	0.0	70144	111745	3216	
142	136013	Status	2	10	3	2	1.0	31136	59964	6164	
168	135428	Photo	1	9	3	10	0.0	41984	68290	3370	

```
In [18]: # concatenating pain==1 or Likes >750
data_frame3=pd.concat([paid_data,likes_750])
data_frame3.head()
```

Out[18]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetime Engaged Users	Co
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	
6	139441	Photo	3	12	1	3	1.0	11692	19479	481	
7	139441	Photo	3	12	7	9	1.0	13720	24137	537	
14	138414	Photo	2	12	4	5	1.0	22784	39941	887	
17	138414	Photo	1	12	2	12	1.0	53264	111785	1706	

```
In [19]: #sort data according to number of comment
data.sort_values(by=['comment']).head()
```

Out[19]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	Lifetime Post Total Reach	Lifetime Post Total Impressions	Lifetim Engage User
499	81370	Photo	2	1	4	4	0.278557	4188	7292	56
111	136736	Photo	1	10	6	8	0.000000	1261	2158	3
276	126424	Photo	3	7	3	4	1.000000	2316	3611	72
282	126345	Status	2	6	7	11	0.000000	12044	20327	224
286	126141	Photo	1	6	5	11	1.000000	2431	4180	38

```
In [20]: transpose=data.transpose()
transpose.head()
```

Out[20]:

	0	1	2	3	4	5	6	7	8	9	...
Page total likes	139441	139441	139441	139441	139441	139441	139441	139441	139441	139441	...
Type	Photo	Status	Photo	Photo	Photo	Status	Photo	Photo	Status	Photo	...
Category	2	2	3	2	2	2	3	3	2	3	...
Post Month	12	12	12	12	12	12	12	12	12	12	...
Post Weekday	4	3	3	2	2	1	1	7	7	6	...

5 rows × 500 columns

```
In [21]: data.shape
```

Out[21]: (500, 19)

```
In [27]: # Reshaping the data
frame1 = data.pivot_table(index = 'Type', columns = 'Paid', values = 'share')
frame1
```

Out[27]:

	Paid	0.000000	0.278557	1.000000
Type				
Link	13.875000		NaN	10.333333
Photo	25.353152		28.0	31.798319
Status	26.542857		NaN	48.500000
Video	59.333333		NaN	46.750000

```
In [28]: frame1 = data.pivot_table(index = ['like', 'Type'], columns = 'Paid', values = frame1
```

```
Out[28]:
```

		Paid	0.000000	0.278557	1.000000
	like	Type			
	0.0	Photo	0.000000	NaN	0.0
	1.0	Photo	2.000000	NaN	NaN
	2.0	Photo	9.088710	NaN	NaN
	3.0	Photo	0.666667	NaN	NaN
	4.0	Photo	0.666667	NaN	2.0

	1572.0	Photo	NaN	NaN	147.0
	1622.0	Photo	208.000000	NaN	NaN
	1639.0	Photo	122.000000	NaN	NaN
	1998.0	Photo	NaN	NaN	128.0
	5172.0	Photo	NaN	NaN	790.0

303 rows × 3 columns

```
In [31]: frame1.unstack()
```

```
Out[31]:
```

		Paid		0.000000		0.278557		1.000000					
	Type	Link	Photo	Status	Video	Link	Photo	Status	Video	Link	Photo	Status	Video
	like												
	0.0	NaN	0.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN
	1.0	NaN	2.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	2.0	NaN	9.088710	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	3.0	NaN	0.666667	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	4.0	NaN	0.666667	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0	NaN	NaN

	1572.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	147.0	NaN	NaN
	1622.0	NaN	208.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	1639.0	NaN	122.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	1998.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	128.0	NaN	NaN
	5172.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	790.0	NaN	NaN

258 rows × 12 columns



In [32]:

frame1.unstack(0)

Out[32]:

	Paid											0.0	...
like	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	...	1047.0	11!
Type													
Link	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	...	NaN	↑
Photo	0.0	2.0	9.08871	0.666667	0.666667	NaN	1.5	6.044355	2.0	1.0	...	98.0	↑
Status	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	↑
Video	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	↑

4 rows × 774 columns

In [33]:

frame1.unstack(1)

Out[33]:

	Paid				0.000000				0.278557				1.000000			
Type	Link	Photo	Status	Video	Link	Photo	Status	Video	Link	Photo	Status	Video	Link	Photo	Status	Video
like																
0.0	NaN	0.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN			
1.0	NaN	2.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
2.0	NaN	9.088710	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
3.0	NaN	0.666667	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
4.0	NaN	0.666667	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0	NaN	NaN			
...			
1572.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	147.0	NaN	NaN			
1622.0	NaN	208.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
1639.0	NaN	122.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN			
1998.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	128.0	NaN	NaN			
5172.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	790.0	NaN	NaN			

258 rows × 12 columns

In []: