ProblemStatement:

Perform the following operations using Python on the Facebook metrics data sets

a. Create data subsets

b. Merge Data

c. Sort Data

d. Transposing Data

e. Shape and reshape Data

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

In [2]:
```
df=pd.read_csv("dataset_Facebook.csv",delimiter=";")
```

In [3]:
```
df
```

Out[3]:

	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
0	139441	Photo	2	12	4	3	0.0	2752	5091	178	109	159
1	139441	Status	2	12	3	10	0.0	10460	19057	1457	1361	1674
2	139441	Photo	3	12	3	3	0.0	2413	4373	177	113	154
3	139441	Photo	2	12	2	10	1.0	50128	87991	2211	790	1119
4	139441	Photo	2	12	2	3	0.0	7244	13594	671	410	580
...	...	...	...	...	...	...	...	...	...	...	...	...
495	85093	Photo	3	1	7	2	0.0	4684	7536	733	708	985
496	81370	Photo	2	1	5	8	0.0	3480	6229	537	508	687
497	81370	Photo	1	1	5	2	0.0	3778	7216	625	572	795
498	81370	Photo	3	1	4	11	0.0	4156	7564	626	574	832
499	81370	Photo	2	1	4	4	NaN	4188	7292	564	524	743

500 rows × 19 columns

In [4]:
```
df.columns
```

Out[4]:
```
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 [7]:
```
df["like"].describe()
```

Out[7]:
```
count     499.000000
mean      177.945892
std       323.398742
min         0.000000
25%        56.500000
50%       101.000000
75%       187.500000
max      5172.000000
Name: like, dtype: float64
```

In [8]:
```
df["Type"].value_counts()
```

Out[8]:
```
Photo     426
Status     45
Link       22
Video       7
Name: Type, dtype: int64
```

In [9]:
```
df["Paid"].value_counts()
```

Out[9]:
```
0.0    360
1.0    139
Name: Paid, dtype: int64
```

a) Subset Of Dataset

In [6]:
```
sub1=df[['Type','Category','Post Month','Post Weekday']].loc[0:15]
```

In [7]:
```
sub1
```

Out[7]:

	Type	Category	Post Month	Post Weekday
0	Photo	2	12	4
1	Status	2	12	3
2	Photo	3	12	3
3	Photo	2	12	2
4	Photo	2	12	2
5	Status	2	12	1
6	Photo	3	12	1
7	Photo	3	12	7
8	Status	2	12	7
9	Photo	3	12	6
10	Status	2	12	5
11	Photo	2	12	5
12	Photo	2	12	5
13	Photo	2	12	5
14	Photo	2	12	4
15	Status	2	12	3

In [9]:
```
sub2=df.loc[0:100]
```

In [10]:
```
sub2.shape
```

Out[10]: (101, 19)

In [13]:
```
sub3=df.loc[101:300]
```

In [14]:
```
sub3.shape
```

Out[14]: (200, 19)

In [17]:
```
sub4=df.loc[0:10,['Type','Post Month']]
sub4
```

Out[17]:

	Type	Post Month
0	Photo	12
1	Status	12
2	Photo	12
3	Photo	12
4	Photo	12
5	Status	12
6	Photo	12
7	Photo	12
8	Status	12
9	Photo	12
10	Status	12

In [18]:
```
sub4.shape
```

Out[18]: (11, 2)

b)Merging of Dataset

Merge Subset by row

In [22]:
```
merge=pd.concat([sub2,sub3])
merge.shape
```

Out[22]: (301, 19)

Merge subset by columns

In [23]:
```
s1=sub2
s1.shape
```

Out[23]: (101, 19)

In [24]:
```
s2=sub3
s2.shape
```

Out[24]: (200, 19)

In [25]:
```
merge2=s1.merge(s2,on='Type')
merge2.shape
```

Out[25]: (12997, 37)

In [26]:
```
st1=df[['Post Month','Type']]
st1.shape
```

Out[26]: (500, 2)

In [27]:
```
st2=df[['Category','Type']]
st2.shape
```

Out[27]: (500, 2)

In [28]:
```
merge3=st1.merge(st2,left_index=True,right_index=True)
merge3.shape
```

Out[28]: (500, 4)

c)Sorting Values

In [20]:
```
sort=df.sort_values('Page total likes',ascending=False)
```

In [21]:
```
sort
```

Out[21]:

	Page total likes	Type	Category	Post Month	Post Weekday	Post Hour	Paid	comment	like	share	Total Interactions
0	139441	Photo	2	12	4	3	0.0	4	79.0	17.0	100
8	139441	Status	2	12	7	3	0.0	0	161.0	31.0	192
1	139441	Status	2	12	3	10	0.0	5	130.0	29.0	164
12	139441	Photo	2	12	5	10	0.0	0	90.0	14.0	104
11	139441	Photo	2	12	5	10	0.0	0	88.0	18.0	106
...	...	...	...	...	...	...	...	...	...	...	...
495	85093	Photo	3	1	7	2	0.0	5	53.0	26.0	84
496	81370	Photo	2	1	5	8	0.0	0	53.0	22.0	75
497	81370	Photo	1	1	5	2	0.0	4	93.0	18.0	115
498	81370	Photo	3	1	4	11	0.0	7	91.0	38.0	136
499	81370	Photo	2	1	4	4	0.0	0	91.0	28.0	119

500 rows × 11 columns

In [29]:
```
sort1=df.sort_values(by='Type')
sort1
```

Out[29]:

	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
438	98195	Link	2	3	6	6	0.0	5730	10083	103	71	97
470	91437	Link	1	2	3	13	0.0	9356	14986	448	381	505
41	138895	Link	1	12	6	3	1.0	18480	28438	517	366	460
43	138353	Link	1	12	5	3	1.0	2645	4270	134	109	170
45	138353	Link	1	12	4	3	1.0	7968	13023	206	158	223
...	...	...	...	...	...	...	...	...	...	...	...	...
71	137893	Video	1	11	5	3	1.0	100768	220447	2101	1735	2331
55	138329	Video	1	11	6	2	1.0	16416	31950	459	411	539
277	126424	Video	1	6	2	13	0.0	139008	277100	1779	1643	2356
243	130791	Video	1	7	3	11	1.0	21872	40413	3872	3822	7327
74	137893	Video	1	11	3	11	0.0	13544	30235	517	458	667

500 rows × 19 columns

Sort the Dataframe based on multiple columns

In [31]:
```
sort2=df.sort_values(by=['Type','Post Month','Paid'])
sort2
```

Out[31]:

	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
477	86909	Link	1	1	6	4	0.0	39600	7927	572	496	581
485	86491	Link	1	1	2	2	0.0	5168	8371	66	59	71
492	85979	Link	1	1	5	11	0.0	45920	5808	753	655	763
481	86491	Link	1	1	4	4	1.0	4938	7910	66	63	70
470	91437	Link	1	2	3	13	0.0	9356	14986	448	381	505
...	...	...	...	...	...	...	...	...	...	...	...	...
183	134879	Video	1	9	2	10	0.0	30624	56950	2080	1956	3253
74	137893	Video	1	11	3	11	0.0	13544	30235	517	458	667
55	138329	Video	1	11	6	2	1.0	16416	31950	459	411	539
71	137893	Video	1	11	5	3	1.0	100768	220447	2101	1735	2331
29	138895	Video	1	12	4	11	1.0	36208	61262	1141	1068	1728

500 rows × 19 columns

d)Transposing Data

In [33]:
```
Tp=df.transpose()
Tp.shape
```

Out[33]: (19, 500)

e)Shape and Reshape Data

In [24]:
```
shaping=df.shape
```

In [25]:
```
shaping
```

Out[25]: (500, 11)

In [27]:
```
reshaping=pd.pivot_table(df,index=['Type','Category'],values='comment')
```

In [28]:
```
print(reshaping)
```

```
                  comment
Type   Category
Link   1          2.900000
       2          2.000000
       3          2.000000
Photo  1          5.897297
       2         11.692308
       3          6.913333
Status 1          4.333333
       2          9.921053
       3          2.750000
Video  1         12.285714
```

In [35]:
```
res=pd.melt(df,id_vars=['Type'],value_vars=['Post Month','Category'])
```

In [36]:
```
res
```

Out[36]:

	Type	variable	value
0	Photo	Post Month	12
1	Status	Post Month	12
2	Photo	Post Month	12
3	Photo	Post Month	12
4	Photo	Post Month	12
...	...	...	...
995	Photo	Category	3
996	Photo	Category	2
997	Photo	Category	1
998	Photo	Category	3
999	Photo	Category	2

1000 rows × 3 columns