In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: data=pd.read_csv(r"C:\Users\user\Downloads\5_Instagram data.csv")
        data
```

Out[2]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits
0	3920	2586	1028	619	56	98	9	5	162	35
1	5394	2727	1838	1174	78	194	7	14	224	48
2	4021	2085	1188	0	533	41	11	1	131	62
3	4528	2700	621	932	73	172	10	7	213	23
4	2518	1704	255	279	37	96	5	4	123	8
...	...	...	...	...	...	...	...	...	...	...
114	13700	5185	3041	5352	77	573	2	38	373	73
115	5731	1923	1368	2266	65	135	4	1	148	20
116	4139	1133	1538	1367	33	36	0	1	92	34
117	32695	11815	3147	17414	170	1095	2	75	549	148

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	I
118	36919	13473	4176	16444	2547	653	5	26	443	611	

119 rows × 13 columns

In [3]:
```
df=data.head(100)
df
```

Out[3]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	
...	...	...	...	...	...	...	...	...	...	...	
95	5394	2275	2975	45	65	61	19	6	147	69	
96	2766	2541	116	51	9	40	10	4	114	11	
97	3924	2244	1278	326	34	139	11	3	151	19	
98	3015	2034	771	115	41	52	11	4	92	9	
99	5409	2643	2006	1068	230	393	10	27	275	38	

100 rows × 13 columns

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     100 non-null    int64 
 1   From Home       100 non-null    int64 
 2   From Hashtags   100 non-null    int64 
 3   From Explore    100 non-null    int64 
 4   From Other      100 non-null    int64 
 5   Saves           100 non-null    int64 
 6   Comments        100 non-null    int64 
 7   Shares          100 non-null    int64 
 8   Likes           100 non-null    int64 
 9   Profile Visits  100 non-null    int64 
 10  Follows         100 non-null    int64 
 11  Caption         100 non-null    object
 12  Hashtags        100 non-null    object
dtypes: int64(11), object(2)
memory usage: 10.3+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

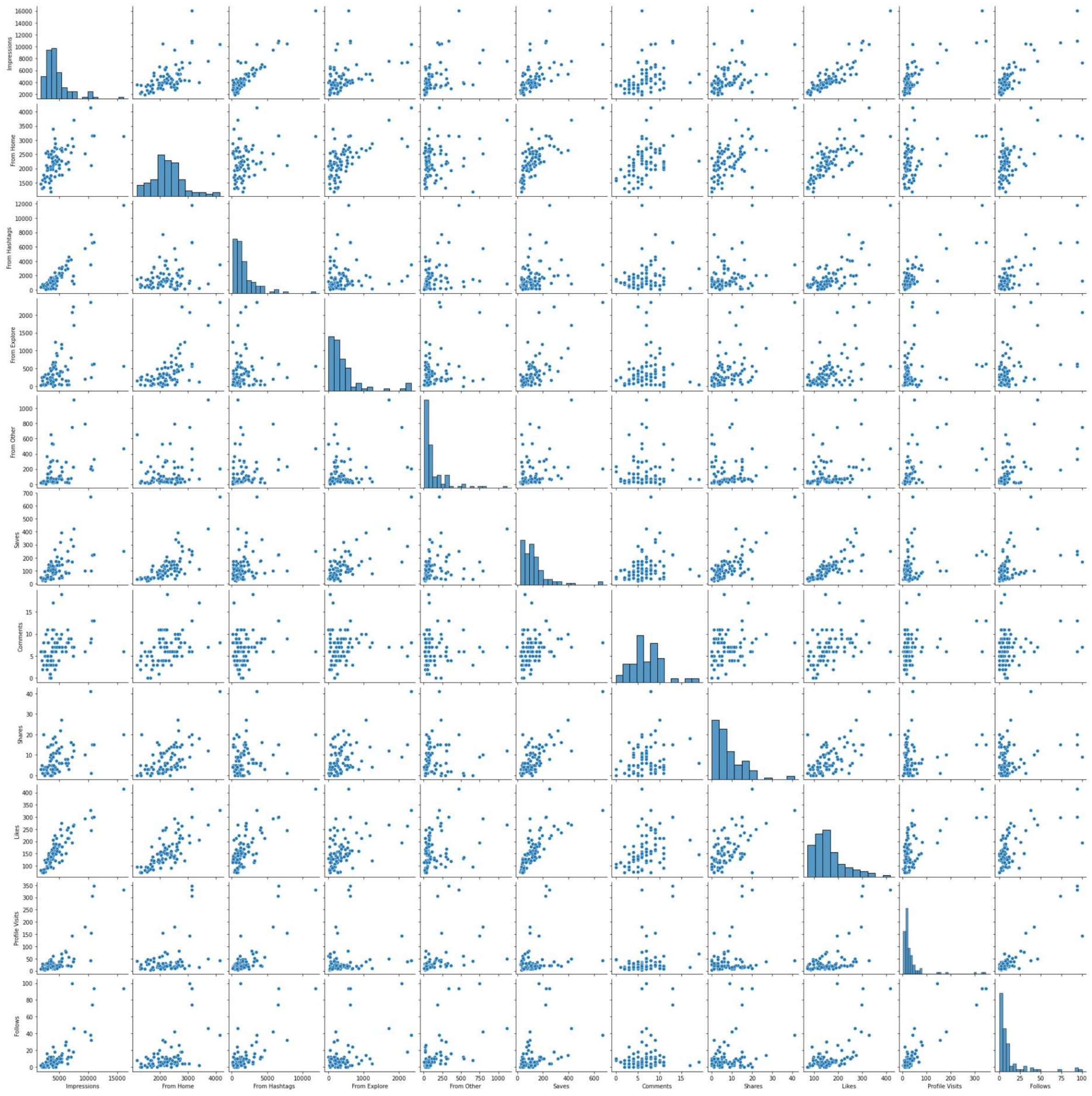
	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Commen
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.0000
mean	4651.780000	2271.560000	1740.930000	423.550000	138.170000	129.570000	6.9900
std	2281.154752	565.495926	1791.644306	513.305182	183.676926	110.526038	3.6055
min	1941.000000	1179.000000	116.000000	0.000000	9.000000	22.000000	0.0000
25%	3229.500000	1967.500000	655.000000	126.250000	36.000000	61.000000	5.0000
50%	3996.000000	2201.000000	1251.000000	249.500000	70.500000	104.500000	7.0000
75%	5303.250000	2588.750000	2208.500000	505.250000	167.250000	145.000000	9.0000
max	16062.000000	4137.000000	11817.000000	2355.000000	1115.000000	668.000000	19.0000

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visit
s',
       'Follows', 'Caption', 'Hashtags'],
      dtype='object')
```

In [7]:
```
sns.pairplot(df)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x1ed52a2ecd0>
```



In [8]:
```
da=df[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
       'Follows', 'Caption', 'Hashtags']]
da
```

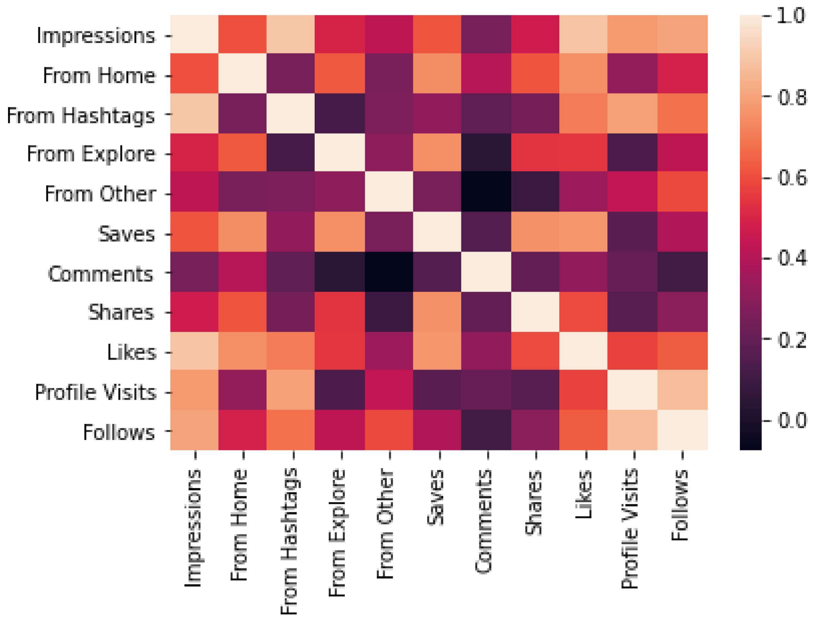
Out[8]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	F
0	3920	2586	1028	619	56	98	9	5	162	35	
1	5394	2727	1838	1174	78	194	7	14	224	48	
2	4021	2085	1188	0	533	41	11	1	131	62	
3	4528	2700	621	932	73	172	10	7	213	23	
4	2518	1704	255	279	37	96	5	4	123	8	
...	...	...	...	...	...	...	...	...	...	...	
95	5394	2275	2975	45	65	61	19	6	147	69	
96	2766	2541	116	51	9	40	10	4	114	11	
97	3924	2244	1278	326	34	139	11	3	151	19	
98	3015	2034	771	115	41	52	11	4	92	9	
99	5409	2643	2006	1068	230	393	10	27	275	38	

100 rows × 13 columns

In [9]:
```
sns.heatmap(da.corr())
```

Out[9]:
```
<AxesSubplot:>
```



In [10]:
```
x=df[['Impressions', 'From Home', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits']]
y=df['Follows']
```

In [11]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [12]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
print(lr.intercept_)
```

```
-11.27996979363559
```

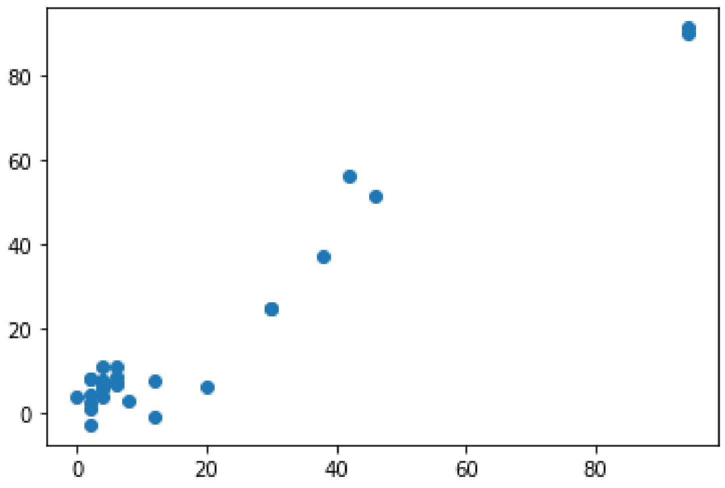
In [14]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[14]:

	Co-efficient
Impressions	0.000976
From Home	0.008235
From Explore	0.006584
From Other	0.021479
Saves	0.000029
Comments	-0.497723
Shares	0.043752
Likes	-0.071429
Profile Visits	0.236238

In [15]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x1ed590bb370>
```



In [16]:
```
print(lr.score(x_test,y_test))
```

```
0.9445970976525906
```

In [17]:
```
print(lr.score(x_train,y_train))
```

```
0.8227543763514334
```

In [18]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [19]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[19]:
```
Ridge(alpha=10)
```

In [20]:
```
rr.score(x_test,y_test)
```

Out[20]:
```
0.9446926868828313
```

In [21]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[21]:
```
Lasso(alpha=10)
```

In [22]:
```
la.score(x_test,y_test)
```

Out[22]:
```
0.943092164442995
```

In [ ]: