In [76]:

```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [472]:

```
x=pd.read_csv(r"C:\Users\user\Downloads\5_Instagram data - 5_Instagram data.cs
```

Out[472]:

	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	
...	...	...	...	...	...	...	...	...	...	...	
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	F
118	36919	13473	4176	16444	2547	653	5	26	443	611	

In [473]:

```
x=x.head(10)
```

Out[473]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Fol
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	
5	3884	2046	1214	329	43	74	7	10	144	9	
6	2621	1543	599	333	25	22	5	1	76	26	
7	3541	2071	628	500	60	135	4	9	124	12	
8	3749	2384	857	248	49	155	6	8	159	36	

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Fol

In [474]:

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     10 non-null     int64 
 1   From Home       10 non-null     int64 
 2   From Hashtags   10 non-null     int64 
 3   From Explore    10 non-null     int64 
 4   From Other      10 non-null     int64 
 5   Saves           10 non-null     int64 
 6   Comments        10 non-null     int64 
 7   Shares          10 non-null     int64 
 8   Likes           10 non-null     int64 
 9   Profile Visits  10 non-null     int64 
 10  Follows         10 non-null     int64 
 11  Caption         10 non-null     object
 12  Hashtags        10 non-null     object
dtypes: int64(11), object(2)
memory usage: 1.1+ KB
```

In [475]:

```
[illegible]
```

Out[475]:

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

In [476]:

```
d=x[['Impressions', 'From Home', 'From Hashtags', 'From Explore']]
```

Out[476]:

	Impressions	From Home	From Hashtags	From Explore
0	3920	2586	1028	619
1	5394	2727	1838	1174
2	4021	2085	1188	0
3	4528	2700	621	932
4	2518	1704	255	279
5	3884	2046	1214	329
6	2621	1543	599	333
7	3541	2071	628	500
8	3749	2384	857	248
9	4115	2609	1104	178

In [477]:

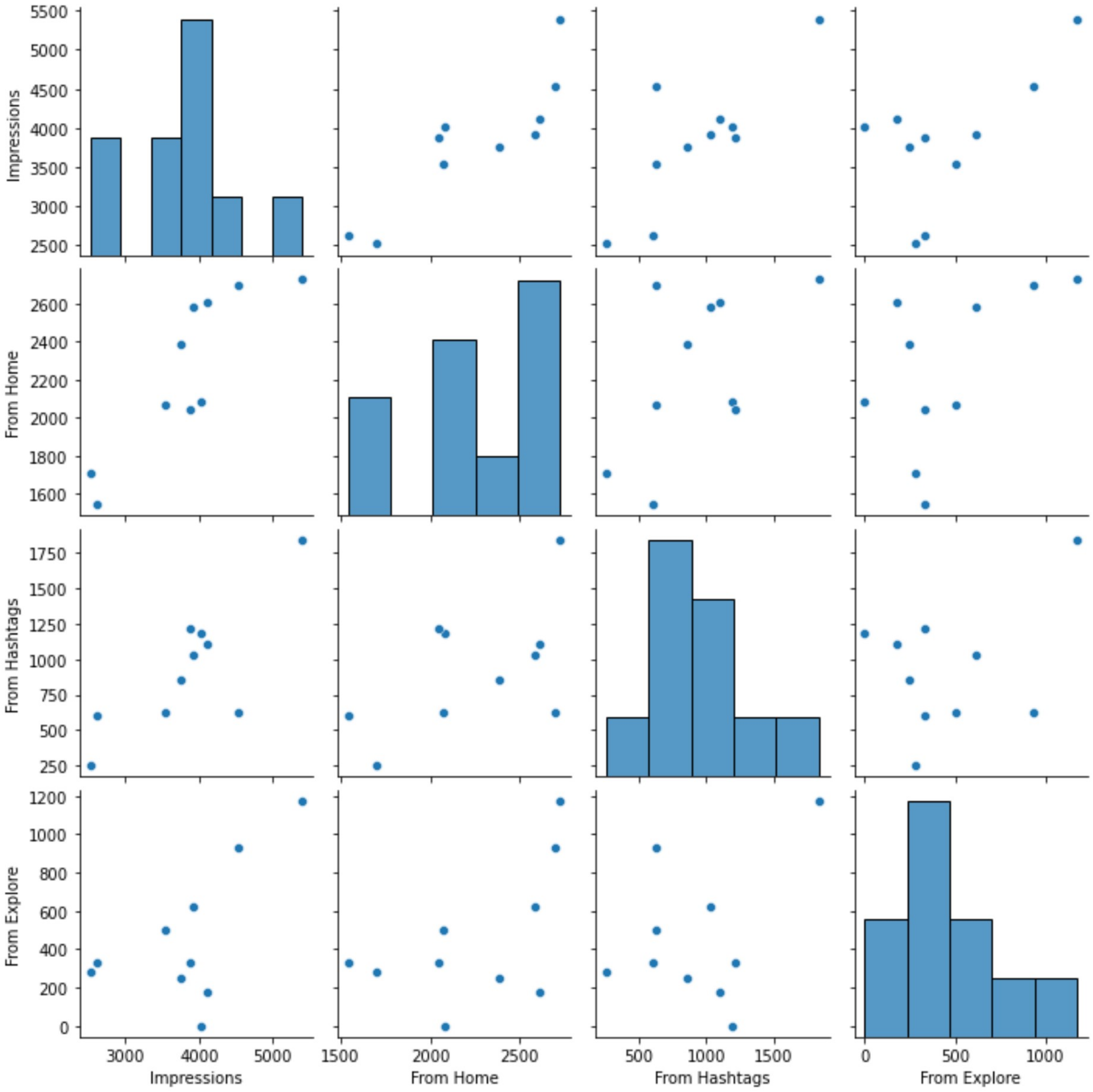
```
[illegible]
```

Out[477]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	3829.100000	2245.500000	933.200000	459.200000	100.000000	110.900000	7.000000
std	838.988869	420.106666	443.303458	359.254413	152.969859	55.604656	2.309401
min	2518.000000	1543.000000	255.000000	0.000000	25.000000	22.000000	4.000000
25%	3593.000000	2052.250000	622.750000	255.750000	43.750000	79.500000	5.250000
50%	3902.000000	2234.500000	942.500000	331.000000	52.500000	110.000000	6.500000
75%	4091.500000	2603.250000	1167.000000	589.250000	69.750000	150.000000	8.500000
max	5394.000000	2727.000000	1838.000000	1174.000000	533.000000	194.000000	11.000000

In [478]:
```
sns.pairplot(d)
```

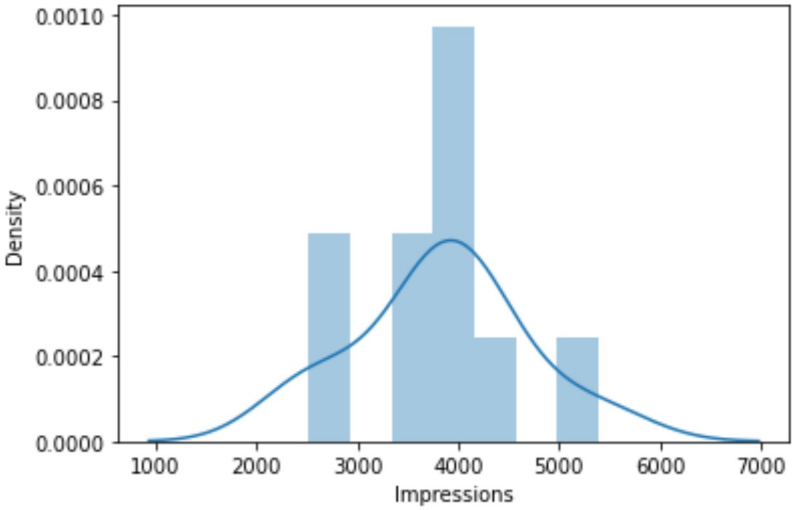
Out[478]:
```
<seaborn.axisgrid.PairGrid at 0x190d3611100>
```



In [479]:
```
sns.distplot(x1['Impressions'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

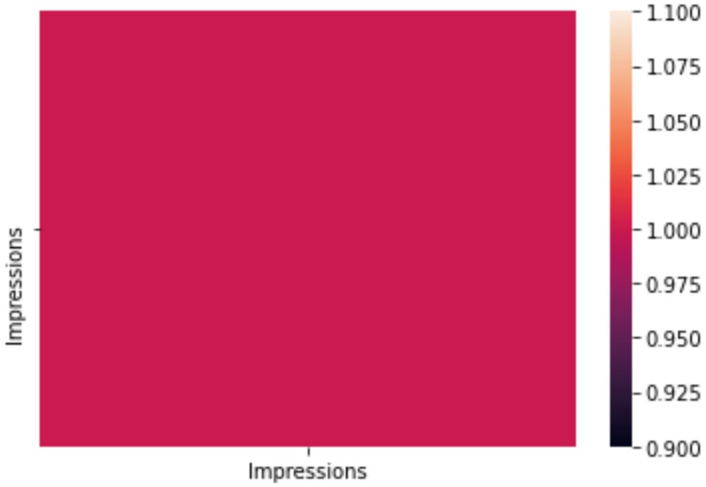
Out[479]:
```
<AxesSubplot:xlabel='Impressions', ylabel='Density'>
```



In [480]:
```
x1=x[['Impressions']]
```

In [481]:
```
sns.heatmap(x1.corr())
```

Out[481]:
```
<AxesSubplot:>
```



In [482]:
```
x=x1[['Impressions']]
y=x1['Impressions']
```

In [483]:

```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [484]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[484]:

```
LinearRegression()
```

In [485]:

```
0.0
```

In [486]:

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

Out[486]:

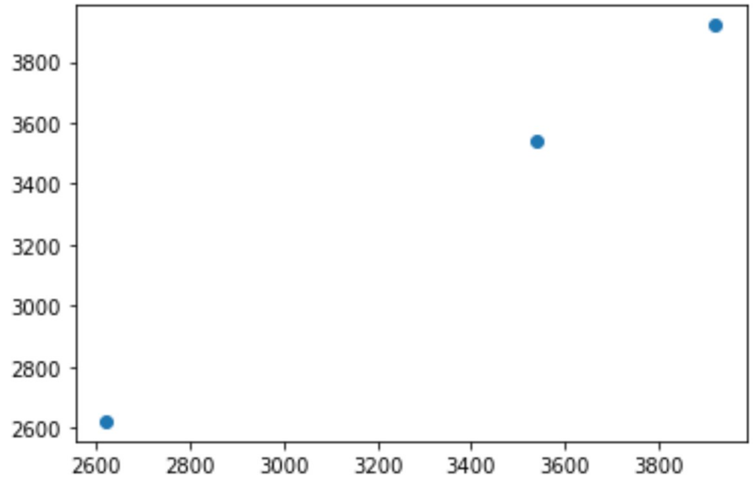
	Co-efficient
Impressions	1.0

In [487]:

```
prediction=lr.predict(x_test)
```

Out[487]:

```
<matplotlib.collections.PathCollection at 0x190d4155f70>
```



In [488]:

Out[488]:

```
1.0
```

In [489]:

Out[489]:

```
1.0
```

In [490]:

```
In [491]: rr=Ridge(alpha=10)
          rr.fit(x_train,y_train)

Out[491]: 0.9999999999876404

In [492]: la=Lasso(alpha=10)

Out[492]: Lasso(alpha=10)

In [493]:

Out[493]: 0.9999999993943798

In [494]: from sklearn.linear_model import ElasticNet
          en=ElasticNet()

Out[494]: ElasticNet()

In [495]:

Out[495]: array([0.99999845])

In [496]:

Out[496]: array([2621.00219047, 3920.0001708 , 3541.00076007])

In [497]:

Out[497]: 0.006265561236432404

In [498]:

Out[498]: 0.9999999999939438

In [499]:

In [500]:

          Mean Absolute Error 0.0

In [501]:

          Mean Squared Error 0.0

In [502]:

          Root Mean Squared Error 0.0

In [ ]:
```