A real estate agent want help to predict the house price for regions in Usa.he gave us the dataset to work on to use linear Regression model.Create a model that helps him to estimate

Data Collection

In [4]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
#import the dataset
data=pd.read_csv(r"C:\Users\user\Desktop\Vicky\5_Instagram data.csv")[0:500]
```

In [6]:
```
#to display top 10 rows
data.head()
```

Out[6]:

	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	

In [7]:
```
#to display null values
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     119 non-null    int64 
 1   From Home       119 non-null    int64 
 2   From Hashtags   119 non-null    int64 
 3   From Explore    119 non-null    int64 
 4   From Other      119 non-null    int64 
 5   Saves           119 non-null    int64 
 6   Comments        119 non-null    int64 
 7   Shares          119 non-null    int64 
 8   Likes           119 non-null    int64 
 9   Profile Visits  119 non-null    int64 
 10  Follows         119 non-null    int64 
 11  Caption         119 non-null    object
 12  Hashtags        119 non-null    object
dtypes: int64(11), object(2)
memory usage: 12.2+ KB
```

In [8]:
```
#to display summary of statistics
data.describe()
```

Out[8]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comn
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.0(
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.6(
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.5(
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.0(
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.0(
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.0(
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.0(
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.0(

In [9]:
```
#to display columns name
data.columns
```

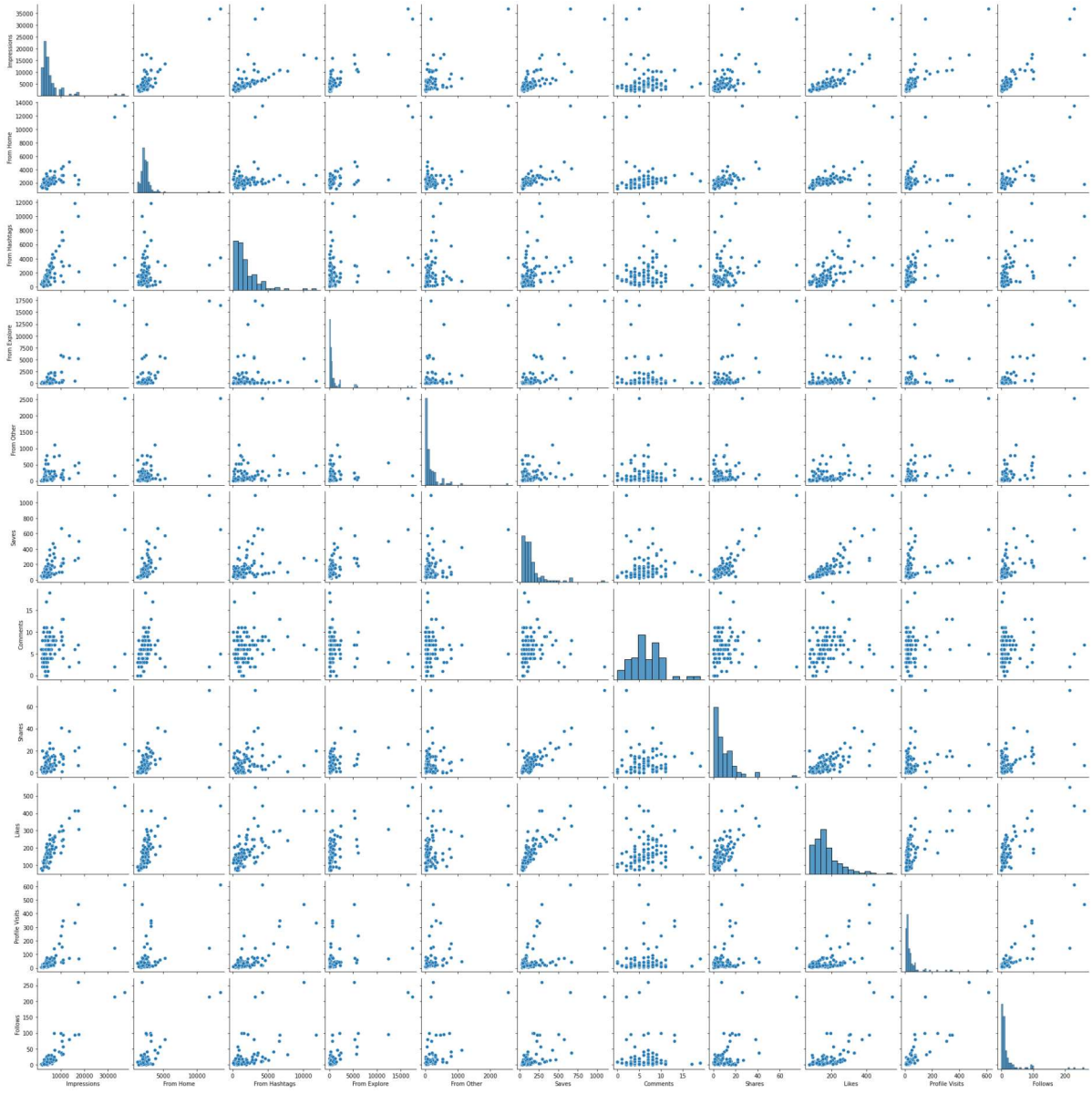
Out[9]:
```
Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visit
s',
       'Follows', 'Caption', 'Hashtags'],
      dtype='object')
```

In [11]:
```
data1=data[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
       'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
       'Follows', 'Caption']]
```

EDA and Visualization

In [12]:
```
sns.pairplot(data1)
```

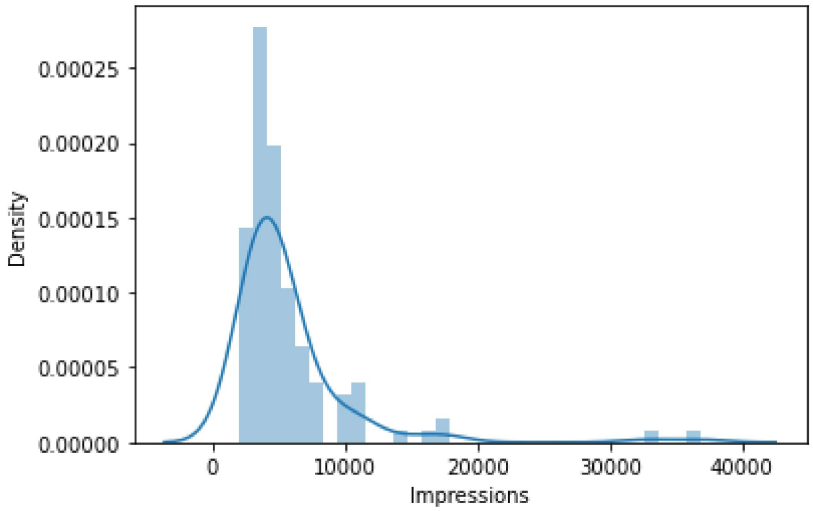
Out[12]:
```
<seaborn.axisgrid.PairGrid at 0x20f9ec3f730>
```



In [14]:
```
sns.distplot(data['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[14]: <AxesSubplot:xlabel='Impressions', ylabel='Density'>

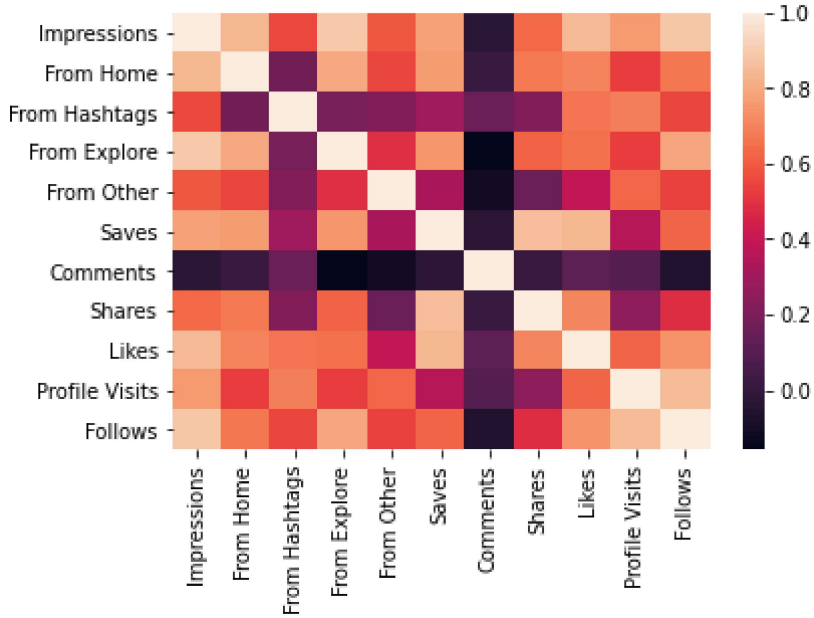


In [ ]:

In [ ]:

In [15]:
```
sns.heatmap(data1.corr())
```

Out[15]: <AxesSubplot:>



To train the model

we are going to train the linear regression model ;We need to split the two variable x and y where x in independent variable (input) and y is dependent of x(output) so we could ignore address columns as it is not requires for our model

In [81]:
```
x=data1[[ 'Comments',  'Likes' ]]
y=data1['From Home']
```

In [82]:
```
#To split test and train data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.6)
```

In [83]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[83]:
```
LinearRegression()
```

In [84]:
```
lr.intercept_
```

Out[84]:
```
488.0135548025951
```

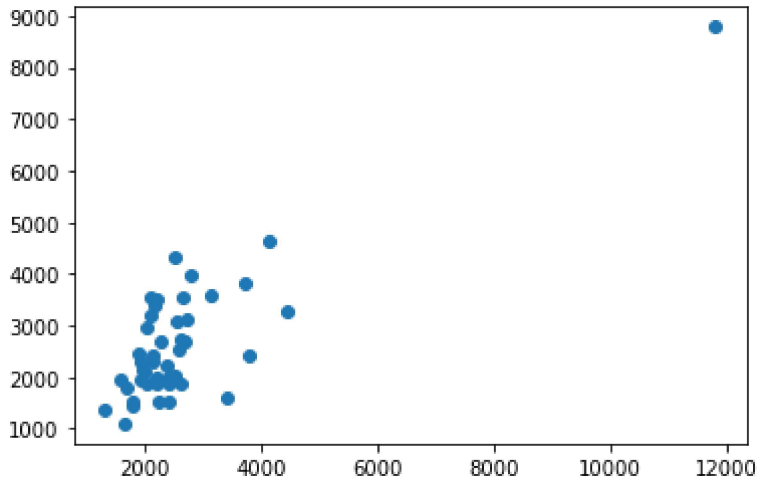
In [85]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[85]:

	Co-efficient
Comments	-121.206490
Likes	15.569582

In [86]:
```
prediction = lr.predict(x_train)
plt.scatter(y_train,prediction)
```

Out[86]:
```
<matplotlib.collections.PathCollection at 0x20fa6ab9190>
```



In [78]:
```
lr.score(x_test,y_test)
```

Out[78]:
```
0.7616618280912317
```

In [ ]:

In [ ]: