

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

```
In [2]: #import dataset
df=pd.read_csv(r"E:\154\5_Instagram data.csv")
df
```

Out[2]:

	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	

119 rows × 13 columns

In [3]: df.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 [4]: #to display top 5 rows
df.head()
```

Out[4]:

	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

Data cleaning and Pre-Processing

```
In [5]: #To find null values
df.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 [6]: # To display summary of statistics
df.describe()
```

```
Out[6]:
```

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments
count	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.666667
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.545455
min	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.000000
25%	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.000000
50%	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.000000
75%	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.000000
max	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.000000

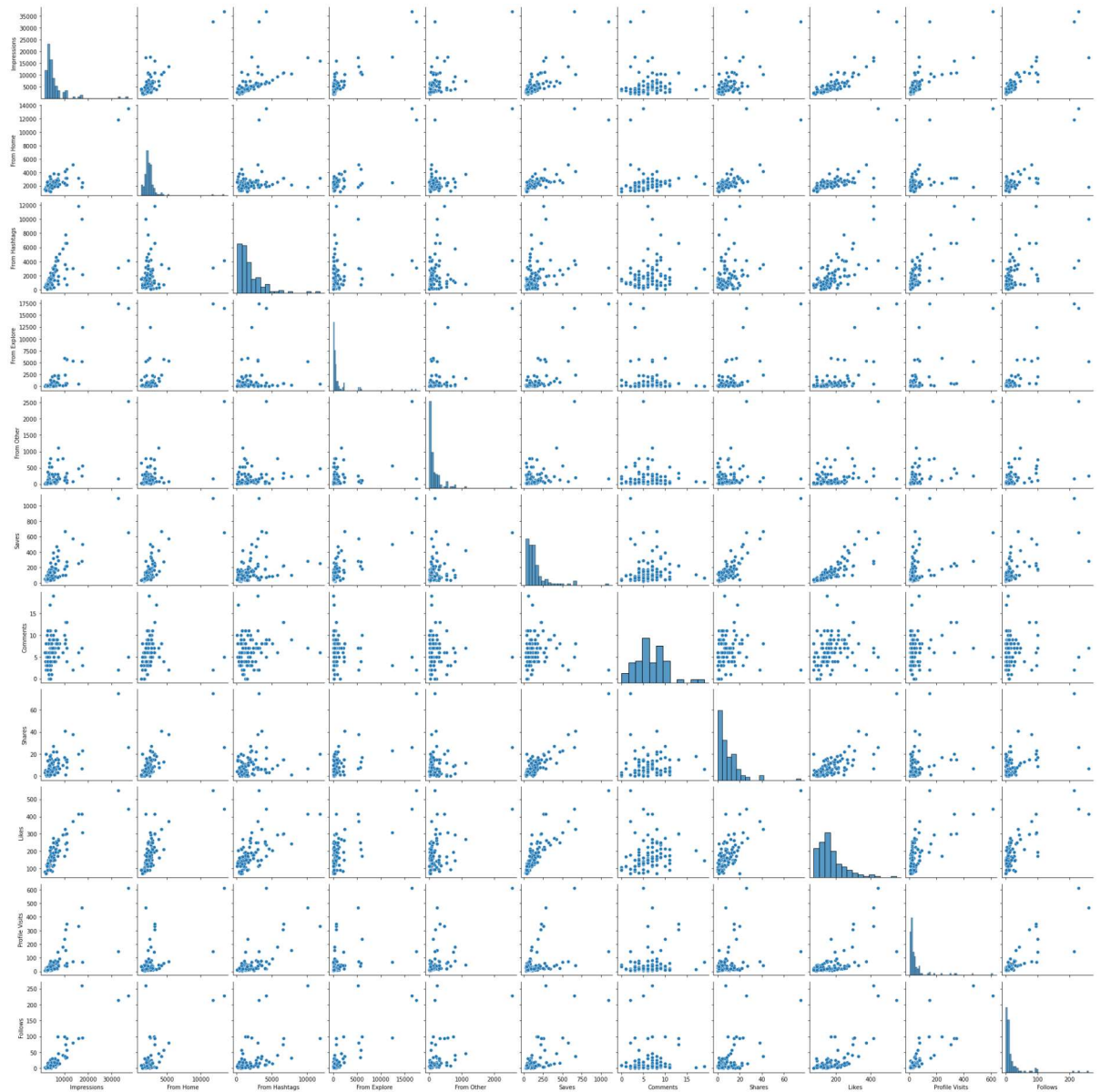
```
In [7]: #To Display column heading
df.columns
```

```
Out[7]: Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',
               'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
               'Follows', 'Caption', 'Hashtags'],
              dtype='object')
```

EDA and VISUALIZATION

```
In [8]: sns.pairplot(df)
```

```
Out[8]: <seaborn.axisgrid.PairGrid at 0x2ad074ae9a0>
```

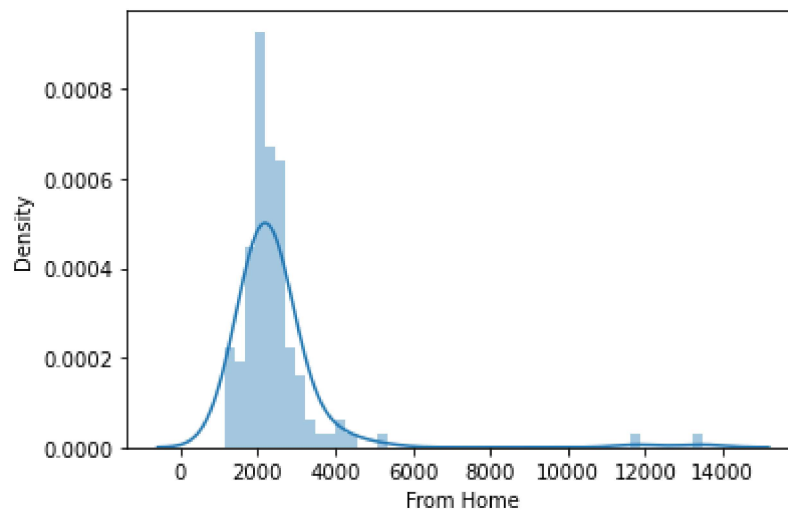


```
In [9]: sns.distplot(df["From Home"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
Out[9]: <AxesSubplot:xlabel='From Home', ylabel='Density'>
```

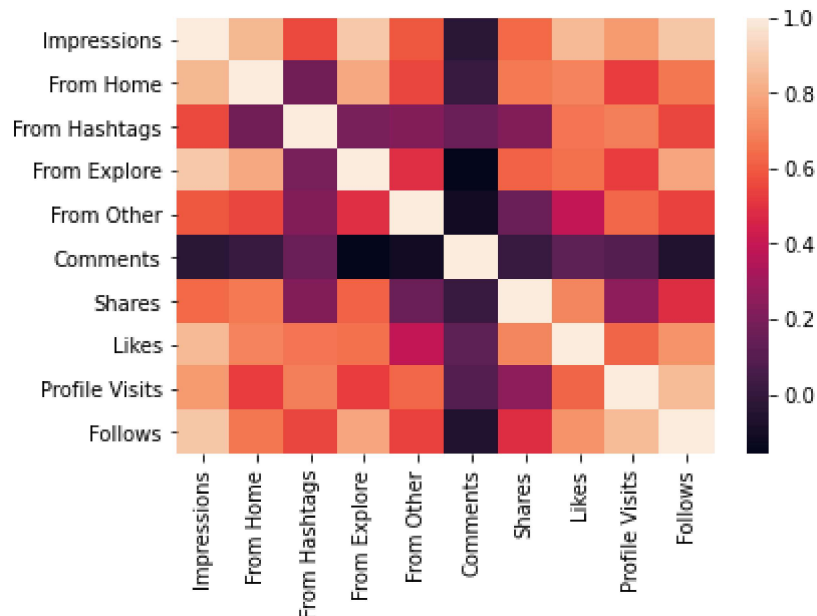


```
In [10]: df1=df[['Impressions', 'From Home', 'From Hashtags', 'From Explore',  
                'From Other', 'Comments', 'Shares', 'Likes', 'Profile Visits',  
                'Follows']]
```


Plot Using Heat Map

```
In [11]: sns.heatmap(df1.corr())
```

```
Out[11]: <AxesSubplot:>
```



To Train The Model-Model Building

we are going to train Linear Regression Model; We need to split out data into two variables x and y where x is independent variable(input) and y is dependent on x(output) we could ignore address column as it required for our model

```
In [14]: x=df1[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
               'From Other', 'Comments', 'Shares', 'Likes', 'Profile Visits',
               'Follows']]
y=df1['Saves']
```

```
-----
KeyError                                Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    3079         try:
-> 3080             return self._engine.get_loc(casted_key)
    3081         except KeyError as err:
```

```
pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()
```

```
pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()
```

```
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

```
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

KeyError: 'Saves'

The above exception was the direct cause of the following exception:

```
KeyError                                Traceback (most recent call last)
<ipython-input-14-98cf027962e1> in <module>
      2         'From Other', 'Comments', 'Shares', 'Likes', 'Profile Visits',
      3         'Follows']]
----> 4 y=df1['Saves']
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    3022         if self.columns.nlevels > 1:
    3023             return self._getitem_multilevel(key)
-> 3024         indexer = self.columns.get_loc(key)
    3025         if is_integer(indexer):
    3026             indexer = [indexer]
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    3080         return self._engine.get_loc(casted_key)
    3081         except KeyError as err:
-> 3082             raise KeyError(key) from err
    3083
    3084         if tolerance is not None:
```

KeyError: 'Saves'

To Split my dataset into training and test data

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

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

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

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

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

Accuracy

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

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

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

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

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

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

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