

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

```
In [170]: #import the dataset
data=pd.read_csv(r"C:\Users\user\Desktop\Vicky\5_Instagram data.csv")
```

In [171]:

#to display top 10 rows  
data.head()

Out[171]:

	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 [172]: *#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 [173]: data.shape

Out[173]: (119, 13)

In [174]: *#to display summary of statistics*  
data.describe()

Out[174]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comn
<b>count</b>	119.000000	119.000000	119.000000	119.000000	119.000000	119.000000	119.00
<b>mean</b>	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.66
<b>std</b>	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.54
<b>min</b>	1941.000000	1133.000000	116.000000	0.000000	9.000000	22.000000	0.00
<b>25%</b>	3467.000000	1945.000000	726.000000	157.500000	38.000000	65.000000	4.00
<b>50%</b>	4289.000000	2207.000000	1278.000000	326.000000	74.000000	109.000000	6.00
<b>75%</b>	6138.000000	2602.500000	2363.500000	689.500000	196.000000	169.000000	8.00
<b>max</b>	36919.000000	13473.000000	11817.000000	17414.000000	2547.000000	1095.000000	19.00

```
In [175]: #to display columns name  
data.columns
```

```
Out[175]: Index(['Impressions', 'From Home', 'From Hashtags', 'From Explore',  
                'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',  
                'Follows', 'Caption', 'Hashtags'],  
               dtype='object')
```

```
In [176]: data1=data[['Impressions', 'From Home', 'From Hashtags', 'From Explore',  
                    'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',  
                    'Follows', 'Caption']]
```

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

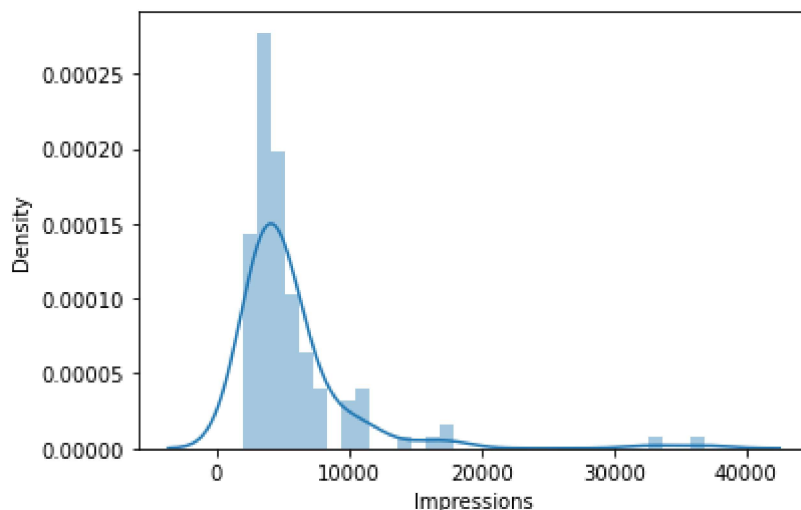
## EDA and Visualization

```
In [179]: sns.distplot(data['Impressions'])
```

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).

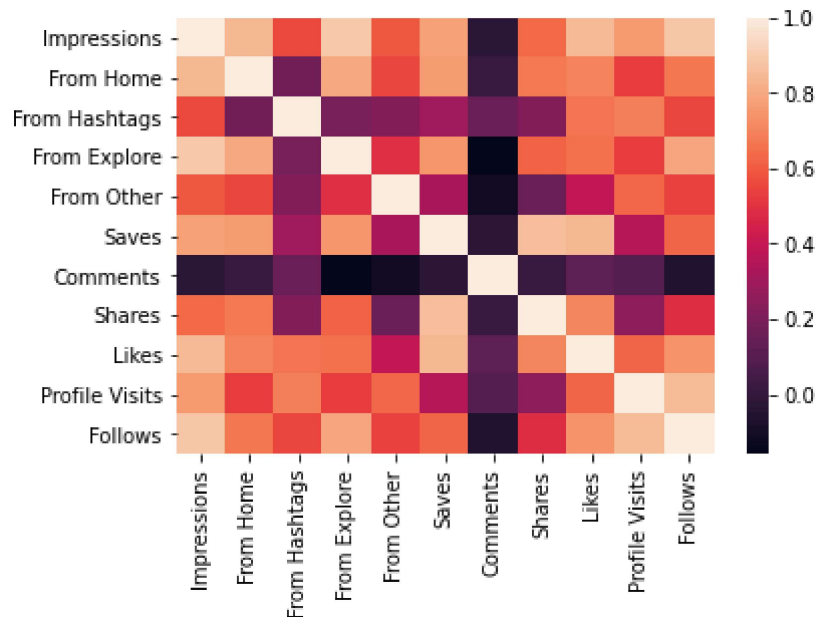
warnings.warn(msg, FutureWarning)

```
Out[179]: <AxesSubplot:xlabel='Impressions', ylabel='Density'>
```



```
In [180]: sns.heatmap(data1.corr())
```

```
Out[180]: <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 [193]: x=data1[[ 'Impressions', 'From Home', 'From Hashtags', 'From Explore',
                  'From Other', 'Saves', 'Comments', 'Shares', 'Likes']]
y=data1["Follows"]
```

```
In [194]: #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 [195]: from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

```
Out[195]: LinearRegression()
```

```
In [196]: lr.intercept_
```

```
Out[196]: -12.300434413054155
```

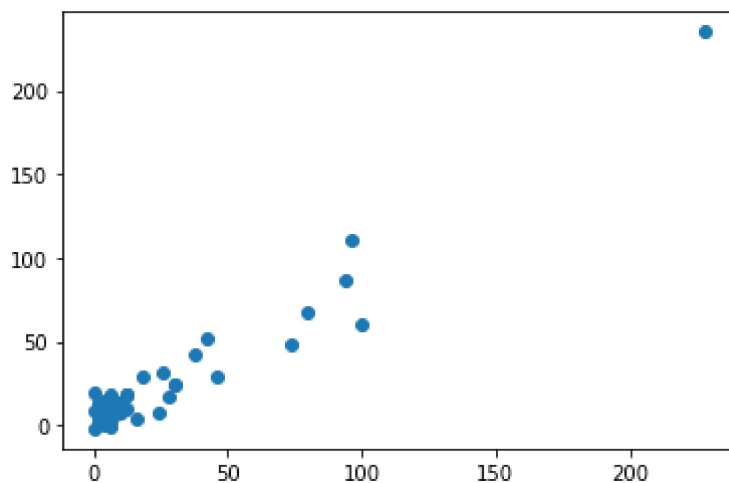
```
In [197]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])  
coeff
```

Out[197]:

	Co-efficient
<b>Impressions</b>	0.008811
<b>From Home</b>	-0.004519
<b>From Hashtags</b>	-0.002504
<b>From Explore</b>	-0.000731
<b>From Other</b>	0.010343
<b>Saves</b>	-0.054430
<b>Comments</b>	-0.052311
<b>Shares</b>	0.719576
<b>Likes</b>	-0.008471

```
In [198]: prediction = lr.predict(x_train)  
plt.scatter(y_train,prediction)
```

Out[198]: <matplotlib.collections.PathCollection at 0x250801767f0>



```
In [199]: lr.score(x_test,y_test)
```

Out[199]: 0.6860251136109589

```
In [200]: lr.score(x_train,y_train)
```

Out[200]: 0.9303997108103291

```
In [201]: from sklearn.linear_model import Ridge,Lasso
```

```
In [202]: rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[202]: 0.6864753966514954

```
In [203]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
la.score(x_test,y_test)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:530: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 1494.9915594844672, tolerance: 7.486774468085107

```
model = cd_fast.enet_coordinate_descent(
```

Out[203]: 0.709572349019992

In [ ]:

In [ ]:

In [ ]:

In [ ]: