

SUMESH R -20104169

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [2]: df = pd.read_csv("13_placement.csv")
df
```

Out[2]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
998	8.62	46.0	1
999	4.90	10.0	1

1000 rows × 3 columns

```
In [3]: df.head()
```

Out[3]:

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows
0	3920	2586	1028	619	56	98	9	5	162	35	2
1	5394	2727	1838	1174	78	194	7	14	224	48	10

	Impressions	From Home	From Hashtags	From Explore	From Other	Saves	Comments	Shares	Likes	Profile Visits	Follows	
												L
2	4021	2085	1188	0	533	41	11	1	131	62	12	r
3	4528	2700	621	932	73	172	10	7	213	23	8	C
												pr
4	2518	1704	255	279	37	96	5	4	123	8	0	an
												v

## Data cleaning and pre processing

In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 3 columns):  
# Column Non-Null Count Dtype  
--- -  
0 cgpa 1000 non-null float64  
1 placement\_exam\_marks 1000 non-null float64  
2 placed 1000 non-null int64  
dtypes: float64(2), int64(1)  
memory usage: 23.6 KB

In [5]: df.describe()

Out[5]:

	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	1
mean	5703.991597	2475.789916	1887.512605	1078.100840	171.092437	153.310924	6.663866	
std	4843.780105	1489.386348	1884.361443	2613.026132	289.431031	156.317731	3.544576	
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 [4]:

```
df.columns
```

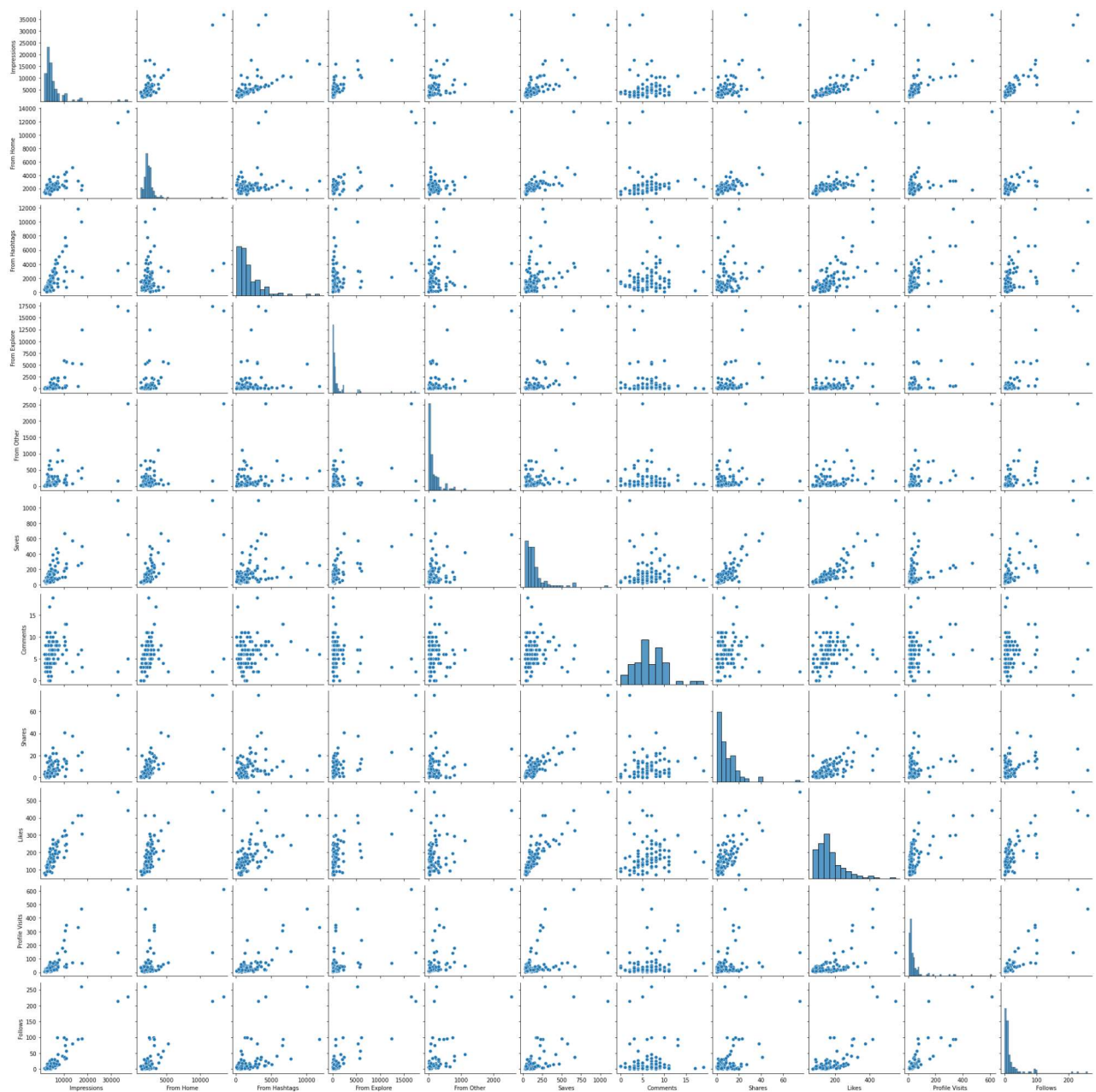
Out[4]: Index(['cgpa', 'placement\_exam\_marks', 'placed'], dtype='object')

# EDA and VISUALIZATION

In [7]:

```
sns.pairplot(df)
```

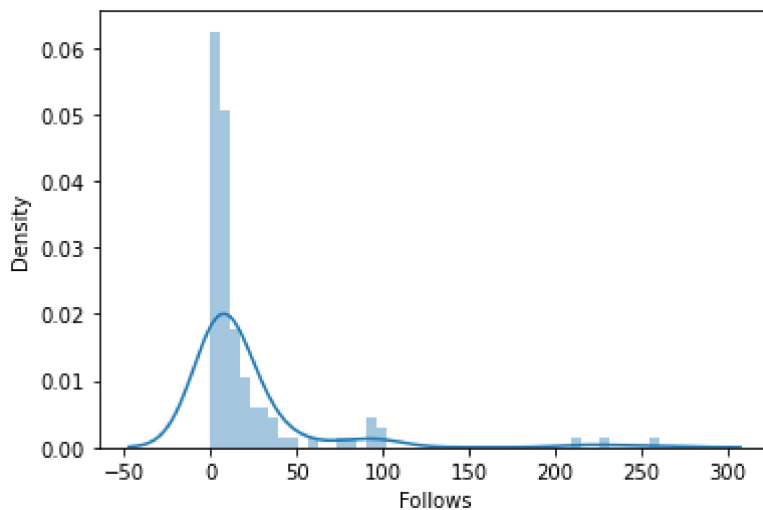
Out[7]: <seaborn.axisgrid.PairGrid at 0x2536e12ea90>



```
In [8]: sns.distplot(df["Follows"])
```

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)

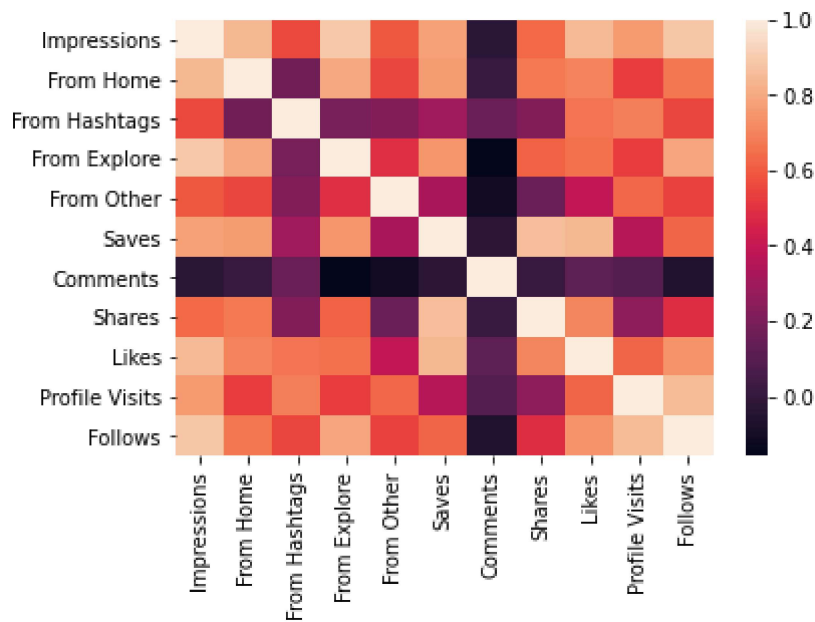
```
Out[8]: <AxesSubplot:xlabel='Follows', ylabel='Density'>
```



```
In [9]: df1 = df[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
                'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits',
                'Follows', 'Caption', 'Hashtags']]
```

```
In [10]: sns.heatmap(df1.corr())
```

Out[10]: <AxesSubplot:>



```
In [11]: x = df1[['Impressions', 'From Home', 'From Hashtags', 'From Explore',
                'From Other', 'Saves', 'Comments', 'Shares', 'Likes', 'Profile Visits']]
y = df1['Follows']
```

## split the data into training and test data

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

```
In [13]: lr = LinearRegression()  
lr.fit(x_train, y_train)
```

```
Out[13]: LinearRegression()
```

```
In [14]: lr.intercept_
```

```
Out[14]: -2.445139943798008
```

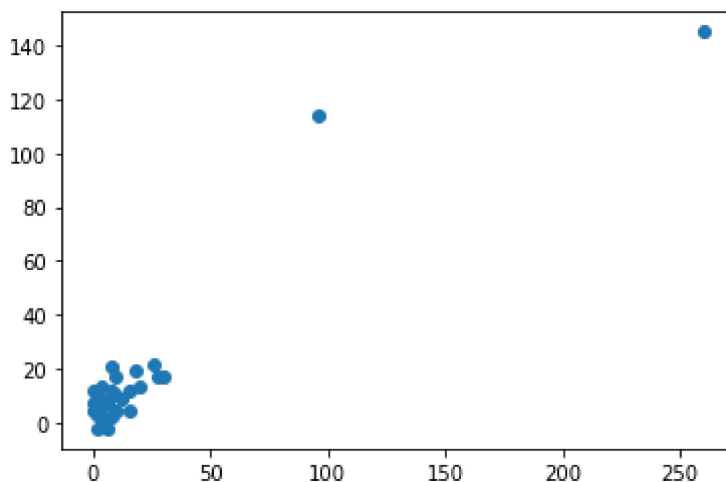
```
In [15]: coeff = pd.DataFrame(lr.coef_, x.columns, columns=['Co-efficient'])  
coeff
```

```
Out[15]:
```

	Co-efficient
<b>Impressions</b>	0.006917
<b>From Home</b>	-0.007751
<b>From Hashtags</b>	-0.006864
<b>From Explore</b>	-0.000201
<b>From Other</b>	-0.008724
<b>Saves</b>	0.012194
<b>Comments</b>	-0.470552
<b>Shares</b>	0.393468
<b>Likes</b>	0.018053
<b>Profile Visits</b>	0.220435

```
In [16]: prediction = lr.predict(x_test)  
plt.scatter(y_test, prediction)
```

```
Out[16]: <matplotlib.collections.PathCollection at 0x253751e2460>
```



```
In [17]: lr.score(x_test,y_test)
```

```
Out[17]: 0.788861261227931
```