Mutiple Linear Regression

In [1]:
```python
##importing Libraries
```

In [2]:
```python
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [3]:
```python
df = pd.read_csv(r"C:\Users\hp\Documents\Datasets\Marketing_Data.csv")
df.head()
```
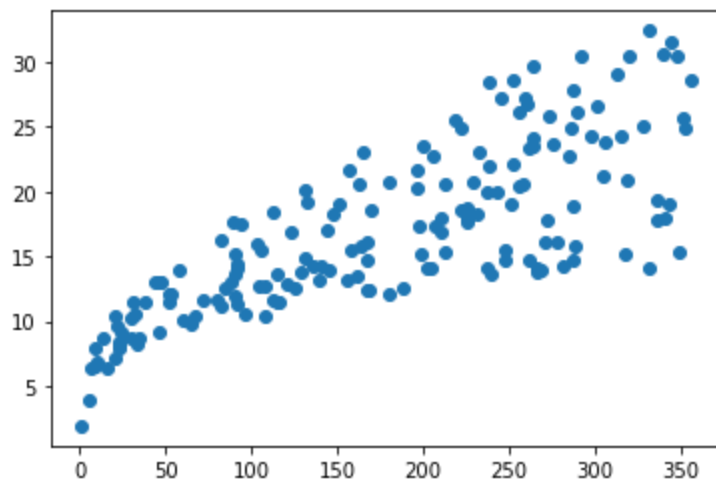
Out[3]:

	youtube	facebook	newspaper	sales
0	84.72	19.20	48.96	12.60
1	351.48	33.96	51.84	25.68
2	135.48	20.88	46.32	14.28
3	116.64	1.80	36.00	11.52
4	318.72	24.00	0.36	20.88

In [4]:
```python
## Data Visualisation
```

In [5]:
```python
plt.scatter(df["youtube"],df["sales"])
```

Out[5]:
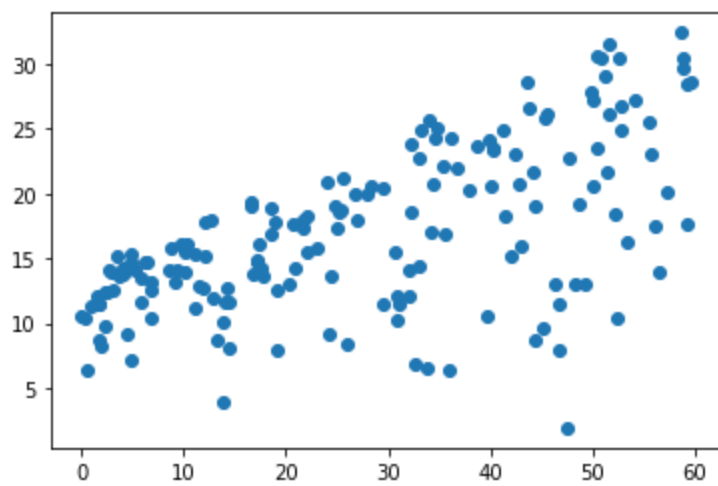```
<matplotlib.collections.PathCollection at 0x2a6e0035820>
```
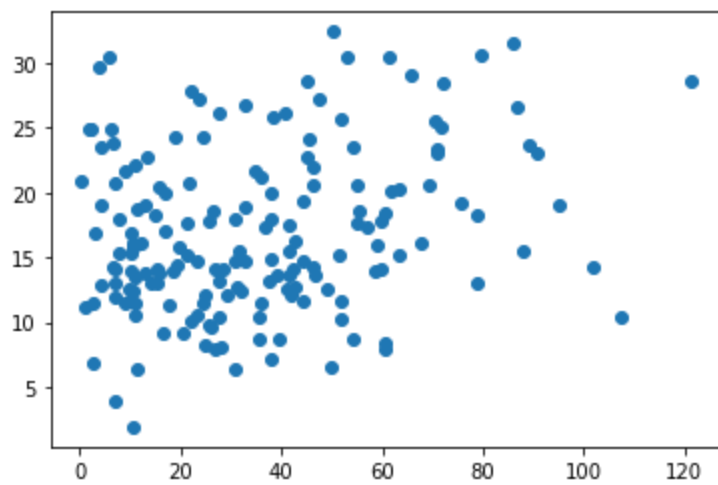


In [6]:
```python
plt.scatter(df["facebook"],df["sales"])
```

Out[6]:
```
<matplotlib.collections.PathCollection at 0x2a6e07f77f0>
```

In [7]:
```python
plt.scatter(df["newspaper"],df["sales"])
```

Out[7]: `<matplotlib.collections.PathCollection at 0x2a6e0863d00>`

In [8]:
```python
## Finding Correlation
```

In [9]:
```python
df["facebook"].corr(df["sales"])
```

Out[9]: `0.6029180163242249`

In [10]:
```python
df["newspaper"].corr(df["sales"])
```

Out[10]: `0.254986978034823`

In [11]:
```python
df["youtube"].corr(df["sales"])
```

Out[11]: `0.7820300203008935`

In [13]:
```python
X = df.drop(["sales"],axis=1)
Y = df["sales"]
```

In [14]:
```python
from sklearn.model_selection import train_test_split
```

```
In [15]:  X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=2)
```

```
In [16]:  reg = LinearRegression()
```

```
In [18]:  model= reg.fit(X_train,Y_train)
          model
```

Out[18]: LinearRegression()

```
In [20]:  model.score(X_train,Y_train)
```

Out[20]: 0.8975490738233135

```
In [21]:  model.predict(X_test)
```

Out[21]: array([20.66257554, 24.59721169, 12.87598477, 24.2923654 , 9.67386865,
                25.70282458, 20.13717958, 16.89843689, 18.73170938, 16.44942632,
                18.31954539, 9.43994383, 20.50908794, 12.85645576, 12.61698551,
                22.99525034, 21.27086474, 13.73061748, 25.97764399, 28.86626428,
                29.45362182, 16.19156588, 23.82399683, 17.09427784, 9.26749032,
                12.62769646, 19.40459335, 20.68904978, 24.64129011, 23.7718537 ,
                17.32803533, 25.46910422, 12.26163395, 14.47926291, 21.84009092])

In [ ]:
```