import relevant libraries

In [14]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

Import data and prepare

In [6]:

```
df=pd.read_csv('D:/NPTEL/BA/Week 4/Tutorial/Hitters1.csv')
df.head()
```

Out[6]:

	Price	Demand
0	3.0	6791
1	3.6	5988
2	40.0	354
3	21.0	651
4	4.0	7111

In [7]:

```
# mark the response variable (y) and the predictor (X)
X=df.iloc[:,0].values
y=df.iloc[:,1].values
X,y
```

Out[7]:

```
(array([ 3. ,  3.6, 40. , 21. ,  4. , 30. , 29. , 18. ,  9. ,  6. ,  5. ,
         8. , 15. , 32. , 20. , 10. , 17. ,  7. , 39. ,  4. , 35. , 39. ,
        44. , 40. , 25. ,  8. ,  5. ,  8. , 10. , 14. , 26. ]),
 array([6791, 5988,  354,  651, 7111,  406,  577,  655, 2891, 5589, 6707,
        4081, 2702,  370, 1251, 3140, 2469, 4909,  396, 7017,  484,  297,
         141,  193, 2801, 3731, 7444, 3910, 2678, 1833, 2946], dtype=int64))
```

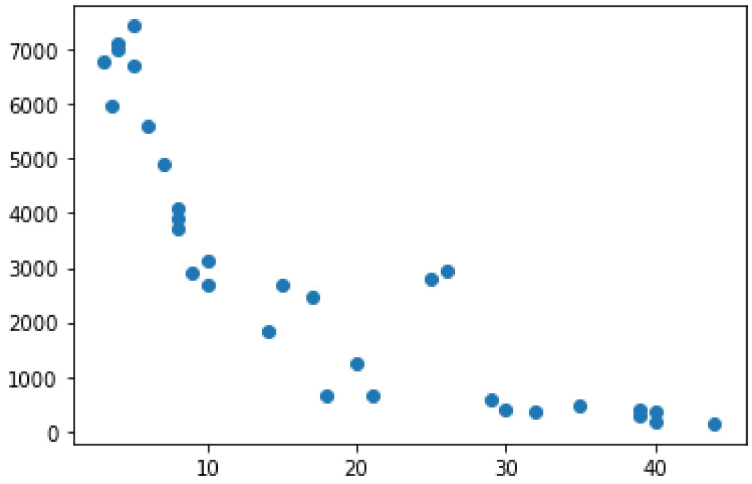
In [8]:

```
# check whether there is a linear pattern between X, y

plt.scatter(X,y)
```

Out[8]:

```
<matplotlib.collections.PathCollection at 0x19799c7a3a0>
```



Fitting a SLR model

In [11]:

```
#reshape your data using array.reshape(-1,1) if your data hase one feature

regressor=LinearRegression()
regressor.fit(X.reshape(-1,1),y)
```

Out[11]:

```
LinearRegression()
```

Output

In [12]:

```
print(regressor.coef_)
print(regressor.intercept_)
```

```
[-157.70087387]
5842.836197653771
```

In [16]:

```
# get the r-squared value
y_pred=regressor.predict(X.reshape(-1,1))
y_pred

r2_score(y,y_pred)
```

Out[16]:

```
0.7338976168337803
```

Transforming data

In [17]:

```
y_new=np.sqrt(y)
y_new
```

Out[17]:

```
array([82.40752393, 77.38216849, 18.81488772, 25.51470164, 84.32674546,
       20.14944168, 24.0208243 , 25.59296778, 53.76802024, 74.75961477,
       81.89627586, 63.88270501, 51.98076567, 19.23538406, 35.36947837,
       56.0357029 , 49.689033  , 70.06425622, 19.89974874, 83.76753548,
       22.        , 17.23368794, 11.87434209, 13.89244399, 52.92447449,
       61.08191222, 86.27861844, 62.5299928 , 51.74939613, 42.81354926,
       54.27706698])
```

In [19]:

```
regressor1=LinearRegression()
regressor1.fit(X.reshape(-1,1),y_new)

print(regressor1.coef_)
print(regressor1.intercept_)

y_pred1=regressor1.predict(X.reshape(-1,1))
y_pred1

r2_score(y_new,y_pred1)
```

```
[-1.68932556]
79.54515270075692
```

Out[19]:

```
0.8308606126447352
```

train test split

In [20]:

```
from sklearn.model_selection import train_test_split
```

In [21]:

```
X_train,X_test,y_train,y_test=train_test_split(X.reshape(-1,1),y,test_size=0.2,random_state
```

In [22]:

```
model_fit=LinearRegression()
model_fit.fit(X_train,y_train)
```

Out[22]:

```
LinearRegression()
```

In [23]:

```
y_pred=model_fit.predict(X_test)
y_pred
```

Out[23]:

```
array([ -742.14288397,  3682.24033676,   619.20579933,  5213.75760547,
        4703.25184923,  4703.25184923, -1422.81722563])
```

In [24]:

```
r2_score(y_test,y_pred)
```

Out[24]:

```
0.6990175696186047
```

In [ ]:

```

```