In [18]:

```
import pandas as pd
```

In [19]:

```
#load the advertising data from the datasets folder
mydata=pd.read_csv(r"C:\Users\cs\Desktop\vaibhav\DataSets-master\Advertising.csv")
```

In [20]:

```
mydata.head()
```

Out[20]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

In [21]:

```
mydata.shape
```

Out[21]:

```
(200, 5)
```

In [22]:

```
#Seperate the data into features and target
X_input=mydata.iloc[0:,1:4]
Y_output=mydata["sales"]
len(X_input)
```

Out[22]:

```
200
```

In [23]:

```
X_input.shape
```

Out[23]:

```
(200, 3)
```

In [24]:

```
X_input.head()
```

Out[24]:

	TV	radio	newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3

	TV	radio	newspaper
3	151.5	41.3	58.5
4	180.8	10.8	58.4

In [25]:

```
#Reshape the data into the numpy to tha Dataframe
ReshapeX=X_input.values.reshape(-1,3)
ReshapeY=Y_output.values.reshape(-1,1)
type(ReshapeX)
```

Out[25]:

```
numpy.ndarray
```

In [26]:

```
#Split the data into train data and test data
X_train=ReshapeX[:140]
X_test=ReshapeX[140:]
Y_train=ReshapeY[:140]
Y_test=ReshapeY[140:]
```

In [56]:

```
X_test.shape
```

Out[56]:

```
(60, 3)
```

In [57]:

```
from sklearn.linear_model import LinearRegression as lg
```

In [58]:

```
#Constructor Calling
teacher=lg()
```

In [59]:

```
#fit(teach) the data into machine
learner=teacher.fit(X_train,Y_train)
```

In [60]:

```
#Predict the answer from the machine
Ya=Y_test
Yalist=list(Y_test)
Yp=learner.predict(X_test)
Yplist=list(Yp)
```

In [61]:

```
#Table between actual value(Ya) and predicted value(Yp)
table=pd.DataFrame({"Ya":Yalist,"Yp":Yplist})
table.head()
```

Out[61]:

	Ya	Yp
0	[10.9]	[9.51435440217]
1	[19.2]	[18.2920287655]
2	[20.1]	[19.2709406424]

	Ya	Yp
2	[20.1]	[19.2709400424]
3	[10.4]	[8.88723581838]
4	[11.4]	[10.1136170553]

In [62]:

```
table.shape
```

Out[62]:

```
(60, 2)
```

In [63]:

```
#find the cofficient values of m
m=learner.coef_
m
```

Out[63]:

```
array([[ 0.04704868,  0.17968299, -0.00300557]])
```

In [64]:

```
m1=m[0,0]
m2=m[0,1]
m3=m[0,2]
```

In [65]:

```
X_test
```

Out[65]:

```
array([[  73.4,   17. ,   12.9],
       [ 193.7,   35.4,   75.6],
       [ 220.5,   33.2,   37.9],
       [ 104.6,    5.7,   34.4],
       [  96.2,   14.8,   38.9],
       [ 140.3,    1.9,    9. ],
       [ 240.1,    7.3,    8.7],
       [ 243.2,   49. ,   44.3],
       [  38. ,   40.3,   11.9],
       [  44.7,   25.8,   20.6],
       [ 280.7,   13.9,   37. ],
       [ 121. ,    8.4,   48.7],
       [ 197.6,   23.3,   14.2],
       [ 171.3,   39.7,   37.7],
       [ 187.8,   21.1,    9.5],
       [   4.1,   11.6,    5.7],
       [  93.9,   43.5,   50.5],
       [ 149.8,    1.3,   24.3],
       [  11.7,   36.9,   45.2],
       [ 131.7,   18.4,   34.6],
       [ 172.5,   18.1,   30.7],
       [  85.7,   35.8,   49.3],
       [ 188.4,   18.1,   25.6],
       [ 163.5,   36.8,    7.4],
       [ 117.2,   14.7,    5.4],
       [ 234.5,    3.4,   84.8],
       [  17.9,   37.6,   21.6],
       [ 206.8,    5.2,   19.4],
       [ 215.4,   23.6,   57.6],
       [ 284.3,   10.6,    6.4],
       [  50. ,   11.6,   18.4],
       [ 164.5,   20.9,   47.4],
       [  19.6,   20.1,   17. ],
       [ 168.4,    7.1,   12.8],
       [ 222.4,    3.4,   13.1],
       [ 276.9,   48.9,   41.8],
       [ 248.4,   30.2,   20.3],
```

```
       [ 170.2,    7.8,   35.2],
       [ 276.7,    2.3,   23.7],
       [ 165.6,   10. ,   17.6],
       [ 156.6,    2.6,    8.3],
       [ 218.5,    5.4,   27.4],
       [  56.2,    5.7,   29.7],
       [ 287.6,   43. ,   71.8],
       [ 253.8,   21.3,   30. ],
       [ 205. ,   45.1,   19.6],
       [ 139.5,    2.1,   26.6],
       [ 191.1,   28.7,   18.2],
       [ 286. ,   13.9,    3.7],
       [  18.7,   12.1,   23.4],
       [  39.5,   41.1,    5.8],
       [  75.5,   10.8,    6. ],
       [  17.2,    4.1,   31.6],
       [ 166.8,   42. ,    3.6],
       [ 149.7,   35.6,    6. ],
       [  38.2,    3.7,   13.8],
       [  94.2,    4.9,    8.1],
       [ 177. ,    9.3,    6.4],
       [ 283.6,   42. ,   66.2],
       [ 232.1,    8.6,    8.7]])
```

In [66]:

```
X_test_tv=X_test[:,0]
X_test_r=X_test[:,1]
X_test_news=X_test[:,2]
len(X_test_tv)
```

Out[66]:

```
60
```

In [67]:

```
len(X_test_news)
```

Out[67]:

```
60
```

In [68]:

```
#values of c
learner.intercept_
```

Out[68]:

```
array([ 3.04514221])
```

In [69]:

```
c=3.045
```

In [70]:

```
#formula mx+c
Yptv=X_test_tv*m1+c
Ypr=X_test_r*m2+c
Ypnews=X_test_news*m3+c
```

In [71]:

```
len(Ypnews)
```

Out[71]:

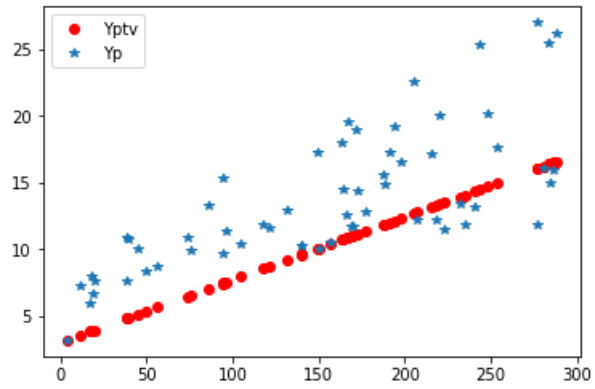
```
60
```

In [72]:

```
#import for data visualization
import matplotlib.pyplot as plt
```

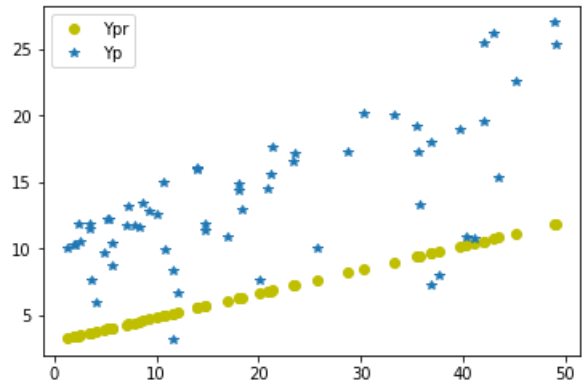
In [73]:

```
#Data visualization of actuaal telivision value and predicted telivision value
plt.plot(X_test_tv,Yptv,"ro")
plt.plot(X_test_tv,Y_test,"*")
plt.legend(["Yptv","Yp"])
plt.show()
```



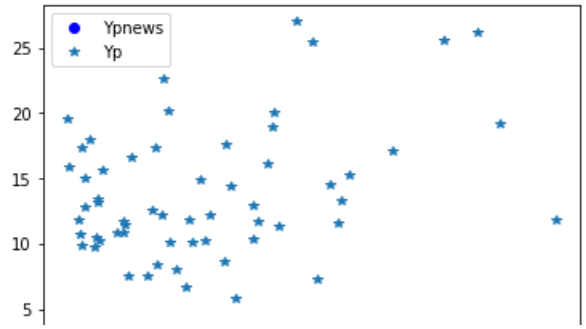
In [74]:

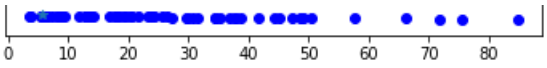
```
#Data visualization of actuaal Radio value and predicted Radio value
plt.plot(X_test_r,Ypr,"yo")
plt.plot(X_test_r,Y_test,"*")
plt.legend(["Ypr","Yp"])
plt.show()
```



In [75]:

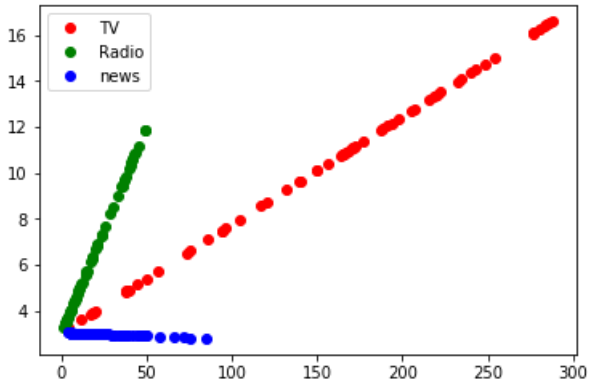
```
#Data visualization of actuaal newspaper value and predicted newspaper value
plt.plot(X_test_news,Ypnews,"bo")
plt.plot(X_test_news,Y_test,"*")
plt.legend(["Ypnews","Yp"])
plt.show()
```





In [76]:

```
#Data visualization of predicted telivision value and predicted Radio value and predicted
Newspaper value
plt.plot(X_test_tv,Yptv,"ro")
plt.plot(X_test_r,Ypr,"go")
plt.plot(X_test_news,Ypnews,"bo")
plt.legend(["TV","Radio","news"])
plt.show()
```



In [77]:

```
#import for finding the mean square error
from sklearn.metrics import mean_squared_error as MSE
import numpy as np
```

In [78]:

```
error=np.sqrt(MSE(Ya,Ypr))
error
```

Out[78]:

```
8.0975557488047496
```

In [79]:

```
error=np.sqrt(MSE(Ya,Yptv))
error
```

Out[79]:

```
4.83097052152808
```

In [80]:

```
error=np.sqrt(MSE(Ya,Ypnews))
error
```

Out[80]:

```
11.913429547760943
```

In [81]:

```
error=np.sqrt(MSE(Ya,Yp))
error
```

Out[81]:

```
1.5993642253910216
```

In [82]:

```
learner.predict([[45,40,69]])
```

Out[82]:

```
array([[ 12.14226841]])
```

In [83]:

```
#pie plot for the data visualization
labels = 'Radio','Television (Max Profit)','Newspaper','Yp'
sizes=[8.097,4.830,11.91,1.599]
explode = (0, 0.07, 0, 0)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',shadow=True, startangle=90)
ax1.axis('equal')
```

Out[83]:

```
(-1.1143878565822356,
 1.1299789263652738,
 -1.1563051022783226,
 1.1026811953465867)
```

In [84]:

```
ax1.set_title("Advertising profit")
plt.show()
```