Sales Prediction System

June 20, 2021

```
[1]: import numpy as np
     import pandas as pd
```

1 Reading the data set

```
[2]: data=pd.read_csv("/home/sumon/Data Science Note/Advertising.csv",index_col=0)
     data
```

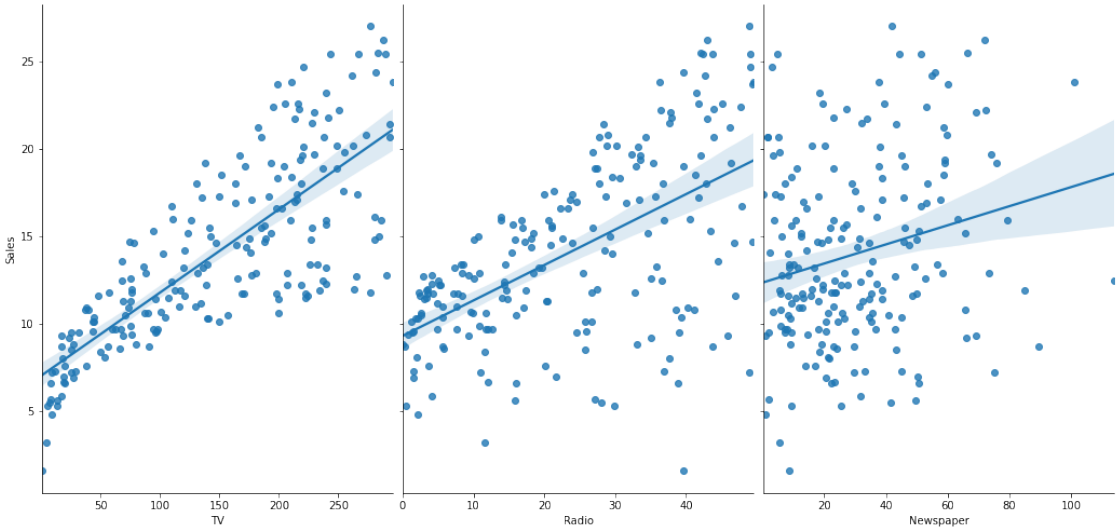
```
[2]:         TV  Radio  Newspaper  Sales
     1    230.1   37.8       69.2   22.1
     2     44.5   39.3       45.1   10.4
     3     17.2   45.9       69.3    9.3
     4    151.5   41.3       58.5   18.5
     5    180.8   10.8       58.4   12.9
     ..     …      …         …      …
     196   38.2    3.7       13.8    7.6
     197   94.2    4.9        8.1    9.7
     198  177.0    9.3        6.4   12.8
     199  283.6   42.0       66.2   25.5
     200  232.1    8.6        8.7   13.4

     [200 rows x 4 columns]
```

2 Visualizing the different columns of the dataset

```
[6]: import seaborn as sb
     sb.pairplot(data,x_vars=['TV','Radio','Newspaper'],y_vars=['Sales'],aspect=0.
      ↪7,height=7,kind='reg')
```

```
[6]: <seaborn.axisgrid.PairGrid at 0x7fcf386f7eb0>
```



3 Correlation matrix

```
[7]: data.corr()
```

```
[7]:                  TV     Radio  Newspaper     Sales
     TV         1.000000  0.054809   0.056648  0.782224
     Radio      0.054809  1.000000   0.354104  0.576223
     Newspaper  0.056648  0.354104   1.000000  0.228299
     Sales      0.782224  0.576223   0.228299  1.000000
```

```
[8]: x=data[['TV','Radio','Newspaper']]
     y=data['Sales']
```

```
[9]: x
```

```
[9]:         TV  Radio  Newspaper
     1    230.1   37.8       69.2
     2     44.5   39.3       45.1
     3     17.2   45.9       69.3
     4    151.5   41.3       58.5
     5    180.8   10.8       58.4
     ..     …      …         …
     196   38.2    3.7       13.8
     197   94.2    4.9        8.1
     198  177.0    9.3        6.4
     199  283.6   42.0       66.2
     200  232.1    8.6        8.7
```

```
[200 rows x 3 columns]
```

```
[10]: y
```

```
[10]: 1      22.1
      2      10.4
      3       9.3
      4      18.5
      5      12.9
             …
      196     7.6
      197     9.7
      198    12.8
      199    25.5
      200    13.4
      Name: Sales, Length: 200, dtype: float64
```

4 Split the data set into training data and testing data

```
[11]: from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
[12]: x_train
```

```
[12]:         TV  Radio  Newspaper
      28   240.1   16.7       22.9
      26   262.9    3.5       19.5
      88   110.7   40.6       63.2
      158  149.8    1.3       24.3
      191   39.5   41.1        5.8
      ..     …      …        …
      132  265.2    2.9       43.0
      170  284.3   10.6        6.4
      45    25.1   25.7       43.3
      59   210.8   49.6       37.7
      131    0.7   39.6        8.7

      [140 rows x 3 columns]
```

```
[13]: y_train
```

```
[13]: 28     15.9
      26     12.0
      88     16.0
      158    10.1
      191    10.8
             …
```

```
132    12.7
170    15.0
45      8.5
59     23.8
131     1.6
Name: Sales, Length: 140, dtype: float64
```

[14]:

```
x_test
```

[14]:

```
        TV  Radio  Newspaper
69   237.4   27.5       11.0
151  280.7   13.9       37.0
46   175.1   22.5       31.5
15   204.1   32.9       46.0
70   216.8   43.9       27.2
135   36.9   38.6       65.6
169  215.4   23.6       57.6
54   182.6   46.2       58.7
9      8.6    2.1        1.0
34   265.6   20.0        0.3
130   59.6   12.0       43.1
35    95.7    1.4        7.4
93   217.7   33.5       59.0
126   87.2   11.8       25.9
55   262.7   28.8       15.9
199  283.6   42.0       66.2
182  218.5    5.4       27.4
71   199.1   30.6       38.7
129  220.3   49.0        3.2
31   292.9   28.3       43.2
13    23.8   35.1       65.9
110  255.4   26.9        5.5
78   120.5   28.5       14.2
143  220.5   33.2       37.9
118   76.4    0.8       14.8
187  139.5    2.1       26.6
119  125.7   36.9       79.2
192   75.5   10.8        6.0
27   142.9   29.3       12.6
7     57.5   32.8       23.5
134  219.8   33.5       45.1
128   80.2    0.0        9.2
41   202.5   22.3       31.6
44   206.9    8.4       26.4
61    53.5    2.0       21.4
63   239.3   15.5       27.3
89    88.3   25.5       73.4
```

```
163  188.4   18.1        25.6
165  117.2   14.7         5.4
198  177.0    9.3         6.4
77    27.5    1.6        20.7
53   216.4   41.7        39.6
22   237.4    5.1        23.5
48   239.9   41.5        18.5
121  141.3   26.8        46.2
177  248.4   30.2        20.3
73    26.8   33.0        19.3
145   96.2   14.8        38.9
52   100.4    9.6         3.6
174  168.4    7.1        12.8
65   131.1   42.8        28.9
120   19.4   16.0        22.3
85   213.5   43.0        33.8
4    151.5   41.3        58.5
162   85.7   35.8        49.3
82   239.8    4.1        36.9
107   25.0   11.0        29.7
116   75.1   35.0        52.7
146  140.3    1.9         9.0
124  123.1   34.6        12.4
```

[15]:
```
y_test
```

[15]:
```
69     18.9
151    16.1
46     14.9
15     19.0
70     22.3
135    10.8
169    17.1
54     21.2
9       4.8
34     17.4
130     9.7
35      9.5
93     19.4
126    10.6
55     20.2
199    25.5
182    12.2
71     18.3
129    24.7
31     21.4
13      9.2
```

```
110    19.8
78     14.2
143    20.1
118     9.4
187    10.3
119    15.9
192     9.9
27     15.0
7      11.8
134    19.6
128     8.8
41     16.6
44     12.9
61      8.1
63     15.7
89     12.9
163    14.9
165    11.9
198    12.8
77      6.9
53     22.6
22     12.5
48     23.2
121    15.5
177    20.2
73      8.8
145    11.4
52     10.7
174    11.7
65     18.0
120     6.6
85     21.7
4      18.5
162    13.3
82     12.3
107     7.2
116    12.6
146    10.3
124    15.2
Name: Sales, dtype: float64
```

5 Training our model

[17]:

```
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(x_train,y_train)
```

```
[17]: LinearRegression()
```

6 Interpreting model coefficients

```
[18]: print(linreg.coef_)
      print(linreg.intercept_)
```

```
[0.04703762 0.1873887  0.00137635]
2.534126914516472
```

7 Making Predictions

```
[19]: y_pred=linreg.predict(x_test)
      y_pred
```

```
[19]: array([18.86918738, 18.39321512, 15.03001512, 18.36290562, 20.99568379,
              11.59330714, 17.16768147, 19.861346  ,  3.33354307, 18.7755062 ,
               7.64555405,  7.30815649, 19.13294299,  8.88264154, 20.30958862,
              23.8354359 , 13.86145818, 17.68667618, 22.08296553, 21.67400469,
              10.32166675, 19.59586146, 13.56228235, 19.17939083,  6.29808212,
               9.52600226, 15.4704055 ,  8.11752339, 14.7636339 , 11.41748357,
              19.21259079,  6.31920659, 16.28150586, 13.87661151,  5.45487089,
              16.73232893, 11.56698448, 14.82298479, 10.80898234, 12.61130953,
               4.15597379, 20.58168029, 14.68888488, 21.62054579, 14.26614716,
              19.90535072, 10.00512566,  9.88603872,  9.06059051, 11.80333945,
              16.7607718 ,  6.47556844, 20.68089369, 17.47999607, 13.34162032,
              14.63282951,  5.8122206 , 12.69779014,  9.50193093, 14.8251738 ])
```

```
[20]: y_test
```

```
[20]: 69     18.9
      151    16.1
      46     14.9
      15     19.0
      70     22.3
      135    10.8
      169    17.1
      54     21.2
      9       4.8
      34     17.4
      130     9.7
      35      9.5
      93     19.4
      126    10.6
      55     20.2
      199    25.5
```

```
182    12.2
71     18.3
129    24.7
31     21.4
13      9.2
110    19.8
78     14.2
143    20.1
118     9.4
187    10.3
119    15.9
192     9.9
27     15.0
7      11.8
134    19.6
128     8.8
41     16.6
44     12.9
61      8.1
63     15.7
89     12.9
163    14.9
165    11.9
198    12.8
77      6.9
53     22.6
22     12.5
48     23.2
121    15.5
177    20.2
73      8.8
145    11.4
52     10.7
174    11.7
65     18.0
120     6.6
85     21.7
4      18.5
162    13.3
82     12.3
107     7.2
116    12.6
146    10.3
124    15.2
Name: Sales, dtype: float64
```

[24]:
```
comparison_data=pd.DataFrame()
comparison_data['y_test']=y_test
comparison_data['y_predict']=y_pred
comparison_data
```

[24]:
```
     y_test  y_predict
69     18.9  18.869187
151    16.1  18.393215
46     14.9  15.030015
15     19.0  18.362906
70     22.3  20.995684
135    10.8  11.593307
169    17.1  17.167681
54     21.2  19.861346
9       4.8   3.333543
34     17.4  18.775506
130     9.7   7.645554
35      9.5   7.308156
93     19.4  19.132943
126    10.6   8.882642
55     20.2  20.309589
199    25.5  23.835436
182    12.2  13.861458
71     18.3  17.686676
129    24.7  22.082966
31     21.4  21.674005
13      9.2  10.321667
110    19.8  19.595861
78     14.2  13.562282
143    20.1  19.179391
118     9.4   6.298082
187    10.3   9.526002
119    15.9  15.470406
192     9.9   8.117523
27     15.0  14.763634
7      11.8  11.417484
134    19.6  19.212591
128     8.8   6.319207
41     16.6  16.281506
44     12.9  13.876612
61      8.1   5.454871
63     15.7  16.732329
89     12.9  11.566984
163    14.9  14.822985
165    11.9  10.808982
198    12.8  12.611310
77      6.9   4.155974
```

```
53      22.6  20.581680
22      12.5  14.688885
48      23.2  21.620546
121     15.5  14.266147
177     20.2  19.905351
73       8.8  10.005126
145     11.4   9.886039
52      10.7   9.060591
174     11.7  11.803339
65      18.0  16.760772
120      6.6   6.475568
85      21.7  20.680894
4       18.5  17.479996
162     13.3  13.341620
82      12.3  14.632830
107      7.2   5.812221
116     12.6  12.697790
146     10.3   9.501931
124     15.2  14.825174
```

8 Evaluation metrics of model

[25]:

```python
from sklearn import metrics
```

[26]:

```python
#mean absolute error
m1=metrics.mean_absolute_error(y_test,y_pred)
m1
```

[26]:

```
1.094765870204584
```

[28]:

```python
#mean squared error
m2=metrics.mean_squared_error(y_test,y_pred)
m2
```

[28]:

```
1.8706491944519692
```

[29]:

```python
#root mean squared error
m3=np.sqrt(metrics.mean_squared_error(y_test,y_pred))
m3
```

[29]:

```
1.367716781520198
```

[ ]: