Lab No. 07

Objective: To use Multiple Linear Regression Technique to assess the impact of advertisement on sales increase.

Regression:

- It is a type of supervised learning problem in which the response is continuous. There are two types of regression methods

1. Univariate linear regression
2. Multivariate linear regression

Linear Regression

Linear regression is the MLA that is used to predict the real value using some independent variables given in the dataset. Lineaer regression is a popular technique for following reasons:

1. Fast
2. No Tuning is required
3. Higly Interpretable

Application of regression

Linear regression may be employed to solve problems, such as;

1. House price prediction
2. Tempreture prediction
3. students' CGPA prediction

Functional form of LR

$$y = \beta 0 + \beta 1 * X1 + \beta 2 * X2 + \beta 3 * X3$$

where:

- y is the response
- $\beta 0$ is the intercept
- $\beta 1$ is coefficient for $X1$ (first feature), $\beta 2$ is coefficient for $X2$ and so on

In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
```

In [2]:

```python
data = pd.read_csv('g:/mydata/advertising.csv')
```

In [3]:

```python
data.head(3)
```

Out[3]:

	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

In [4]:

```python
# set unnamed:0 as index using index_col parameter
data = pd.read_csv('g:/mydata/advertising.csv', index_col = 0)
data.tail(2)
```

Out[4]:

	TV	Radio	Newspaper	Sales
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

In [5]:

```python
data.shape
```

```
Out[5]: (200, 4)
```

Features

- TV: Advertising amount spent on Television for a single product in a given market (in thousand dollars)
- Same for Radio and Newspaper

Target

- Sales: sale of a single product in a given market such as market number 7 or 198

In [6]:

```
data['Sales'].shape
```

Out[6]:

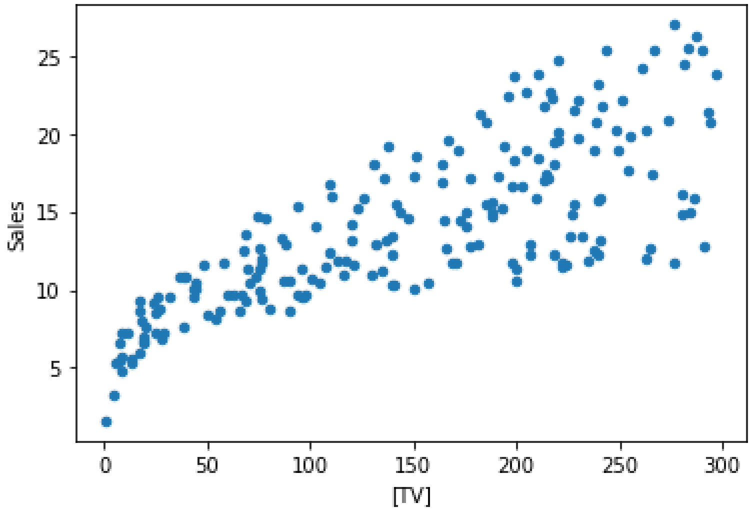
```
(200,)
```

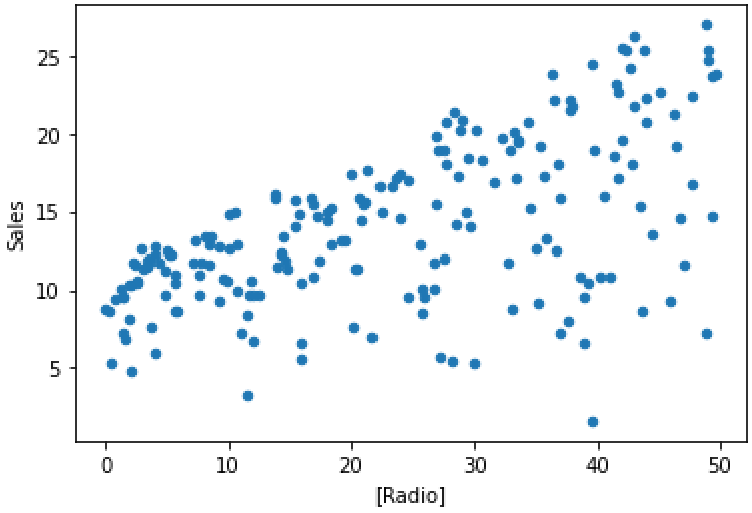
In [7]:

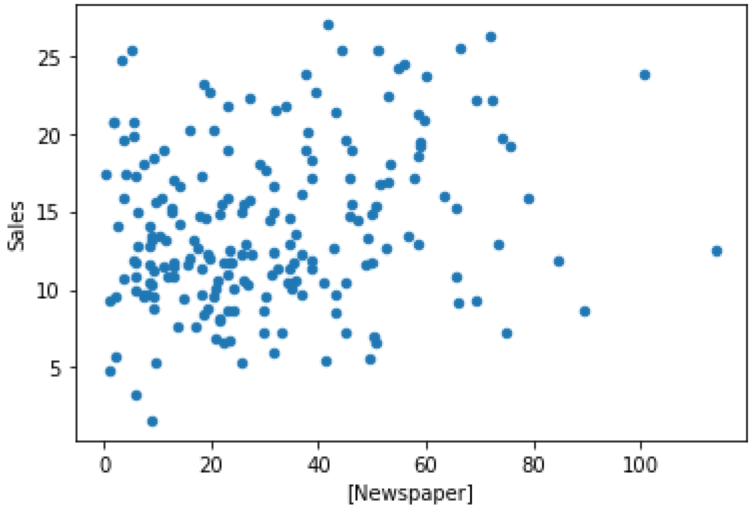
```
data.plot(kind='scatter', x=['TV'], y='Sales')

data.plot(kind='scatter', x=['Radio'], y='Sales')

data.plot(kind='scatter', x=['Newspaper'], y='Sales')
```

Out[7]:

```
<AxesSubplot:xlabel='[Newspaper]', ylabel='Sales'>
```







Concluding results after observing the Graph

1. The relation bw TV and Sales is stong and increases in linear fashion
2. The relation bw Radio and Sales is less stong
3. The relation bw TV and Sales is weak

Split the dataset

In [8]:

```
X = data[['TV', 'Radio', 'Newspaper']]
X.head(2)
```

Out[8]:

	TV	Radio	Newspaper
1	230.1	37.8	69.2
2	44.5	39.3	45.1

In [9]:

```
X.shape
```

Out[9]:

```
(200, 3)
```

In [10]:

```
type(X)
```

Out[10]:

```
pandas.core.frame.DataFrame
```

In [11]:

```
y = data['Sales']
y.head(2)
```

Out[11]:

```
1    22.1
2    10.4
Name: Sales, dtype: float64
```

In [12]:

```
print(y.shape)
```

```
(200,)
```

In [13]:

```
print(type(y))
```

```
<class 'pandas.core.series.Series'>
```

By default

- It splits the given data into 75-25 ration

In [14]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1
```

In [15]:

```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(150, 3)
(150,)
(50, 3)
(50,)
```

In [16]:

```
from sklearn.linear_model import LinearRegression
```

In [17]:

```
lr = LinearRegression()
```

In [18]:

```
lr.fit(X_train, y_train)
```

```
Out[18]: LinearRegression()
```

What Occurs inside the model

In [19]:

```
print(lr.intercept_)
```

```
2.8769666223179318
```

In [20]:

```
print(lr.coef_)
```

```
[0.04656457 0.17915812 0.00345046]
```

In [21]:

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

In [22]:

```
print(y_pred)
```

```
[21.70910292 16.41055243  7.60955058 17.80769552 18.6146359  23.83573998
 16.32488681 13.43225536  9.17173403 17.333853   14.44479482  9.83511973
 17.18797614 16.73086831 15.05529391 15.61434433 12.42541574 17.17716376
 11.08827566 18.00537501  9.28438889 12.98458458  8.79950614 10.42382499
 11.3846456  14.98082512  9.78853268 19.39643187 18.18099936 17.12807566
 21.54670213 14.69809481 16.24641438 12.32114579 19.92422501 15.32498602
 13.88726522 10.03162255 20.93105915  7.44936831  3.64695761  7.22020178
  5.9962782  18.43381853  8.39408045 14.08371047 15.02195699 20.35836418
 20.57036347 19.60636679]
```

Evaluation Metric

In [23]:

```
from sklearn import metrics
```

In [24]:

```
print("MAE:    ", metrics.mean_absolute_error(y_test, y_pred))
print("MSE:    ",metrics.mean_squared_error(y_test, y_pred))
print("RMSE:   ",np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE:     1.0668917082595208
MSE:     1.9730456202283375
RMSE:    1.4046514230328953
```

Result

- The result seems reasonable, given that sales ranges from 5-25

Lab Tasks:

1. Explore the hyper parameters of linear regression algorithm?
2. implement multiple linear regression using support vector regressor algorithm