Linear Regression

In [2]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
dataset=pd.read_csv(r"D:\ML_Course\Works_on_python\Linear_Regression\data.csv")
```

In [4]:
```
dataset.head()
```

Out[4]:

	sno	Temperature	Pressure
0	1	0	0.0002
1	2	20	0.0012
2	3	40	0.0060
3	4	60	0.0300
4	5	80	0.0900

In [5]:
```
dataset.isnull().any()
```

Out[5]:
```
sno            False
Temperature    False
Pressure       False
dtype: bool
```

In [6]:
```
x=dataset.iloc[:,1:2].values
y=dataset.iloc[:,2:3].values
```

In [7]:
```
x.shape
```

Out[7]:
```
(6, 1)
```

In [8]:
```
y.shape
```

Out[8]:
```
(6, 1)
```

In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [10]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[10]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [11]:
```
y_pred=lr.predict(x_test)
```

In [12]:
```
x_test
```

Out[12]:
```
array([[100],
       [ 40]], dtype=int64)
```

In [13]:
```
y_test
```

Out[13]:
```
array([[0.27 ],
       [0.006]])
```

In [14]:
```
y_pred
```

Out[14]:
```
array([[0.09287],
       [0.03035]])
```

In [15]:
```
from sklearn.metrics import r2_score
accuracy=r2_score(y_test,y_pred)
```

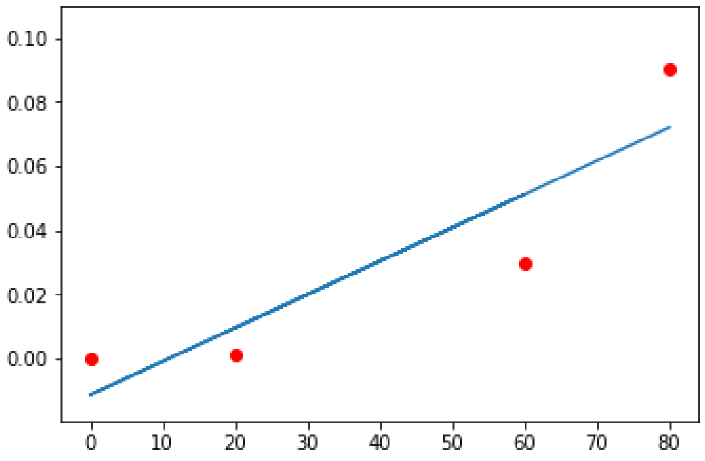
In [16]:
```
accuracy
```

Out[16]:
```
0.08264579315886134
```

In [17]:
```
import matplotlib.pyplot as plt
```

In [18]:
```
plt.scatter(x_train,y_train,color="red")
plt.plot(x_train,lr.predict(x_train))
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x29d15c392e8>]
```

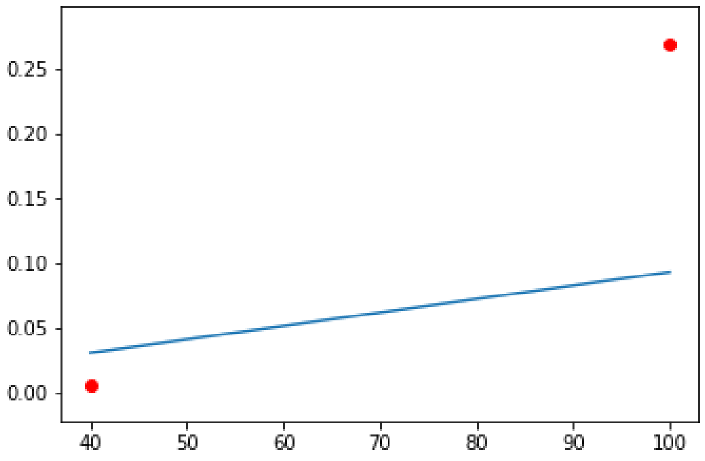


In [19]:

```
plt.scatter(x_test,y_test,color="red")
plt.plot(x_test,y_pred)
```

Out[19]:

```
[<matplotlib.lines.Line2D at 0x29d15d14550>]
```



Decision Tree

In [20]:

```
from sklearn.tree import DecisionTreeRegressor
dtr=DecisionTreeRegressor(random_state=0)
dtr.fit(x_train,y_train)
```

Out[20]:

```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=0, splitter='best')
```

In [21]:

```
y_dtr=dtr.predict(x_test)
```

In [22]:

```
accuracy_dtr=r2_score(y_test,y_dtr)
```

In [24]:

```
accuracy_dtr
```

Out[24]:

```
0.06958677685950398
```

Random Forest

In [62]:

```
from sklearn.ensemble import RandomForestRegressor
rfr=RandomForestRegressor(n_estimators=4,random_state=0)
rfr.fit(x_train,y_train)
```

```
C:\Users\anikp\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a
1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[62]:

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=4, n_jobs=None,
           oob_score=False, random_state=0, verbose=0, warm_start=False)
```

In [63]:

```
y_rfr=rfr.predict(x_test)
```

In [64]:

```
accuracy_rfr=r2_score(y_test,y_rfr)
```

In [65]:

```
accuracy_rfr
```

Out[65]:

```
0.0700826446280991
```

In [ ]: