P Sai Deekshith

1BM18CS148

Multiple Linear Regression

Dataset: ¶

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

In [34]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
%matplotlib inline
```

ML-Academics/ | 1BM18CS148_PSaiDee | Inbox (3,068) - deeksh | Inbox (242) - saidekshi | Writeup - Google Driv | B4 batch ML_LAB TEST | Meet - vxc-qqvd-i

localhost:8888/notebooks/ML-Academics/1BM18CS148_PSaiDeekshith_ML_Lab_CIE2.ipynb#Training

jupyter 1BM18CS148_PSaiDeekshith_ML_Lab_CIE2 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

In [35]:
```
df = pd.read_csv(r"D:\6th Sem\Machine Learning\LAB CIE 2\dataset.csv")
```

In [36]:
```
df.head()
```

Out[36]:

	Height	Age	Weight
0	5.00	45	77
1	5.11	26	47
2	5.60	30	55
3	5.90	34	59
4	4.80	40	72

Splitting the dataset into test and train

In [37]:
```
X = df[['Age', 'Height']]
y = df[['Weight']]
```

In [38]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [39]:
```
X_train
```

Out[39]:

	Age	Height
2	30	5.60
1	26	5.11
0	45	5.00
8	23	5.50
7	28	5.80
4	40	4.80
9	32	5.60

Training

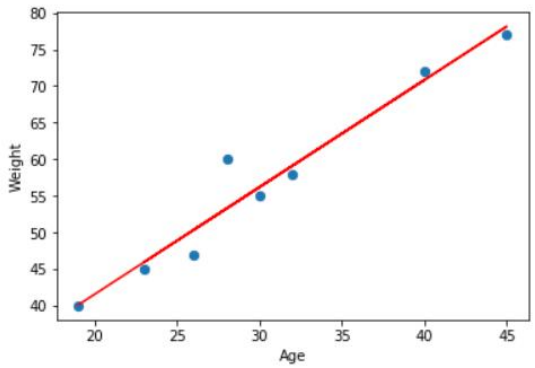
In [47]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[47]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Visualizations

In [60]:
```
#Visualization
reg = LinearRegression()
reg.fit(X_train[['Age']], y_train.Weight)
plt.scatter(X_train.Age,y_train.Weight,s=40)
plt.plot(X_train.Age, reg.predict(X_train[['Age']]), color='red')
plt.xlabel('Age')
plt.ylabel('Weight')
```

Out[60]:
```
Text(0, 0.5, 'Weight')
```



In [62]:
```
#Visualization
reg1 = LinearRegression()
```

ML-Academics/ | 1BM18CS148_PSaiDee | Inbox (3,068) - deeksh | Inbox (242) - saidekshi | Writeup - Google Driv | B4 batch ML_LAB TES | Meet - vxc-qqvd-t

localhost:8888/notebooks/ML-Academics/1BM18CS148_PSaiDeekshith_ML_Lab_CIE2.ipynb#Training

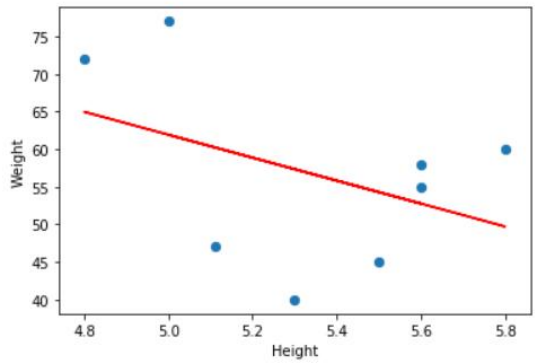
jupyter 1BM18CS148_PSaiDeekshith_ML_Lab_CIE2 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

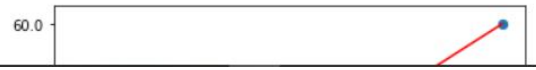
In [62]:
```
#Visualization
reg1 = LinearRegression()
reg1.fit(X_train[['Height']], y_train.Weight)
plt.scatter(X_train.Height,y_train.Weight,s=40)
plt.plot(X_train.Height, reg1.predict(X_train[['Height']]), color='red')
plt.xlabel('Height')
plt.ylabel('Weight')
```

Out[62]: Text(0, 0.5, 'Weight')



In [65]:
```
#Visualization
reg2 = LinearRegression()
reg2.fit(X_test[['Age']], y_test.Weight)
plt.scatter(X_test.Age,y_test.Weight,s=40)
plt.plot(X_test.Age, reg2.predict(X_test[['Age']]), color='red')
plt.xlabel('Age')
plt.ylabel('Weight')
```

Out[65]: Text(0, 0.5, 'Weight')



ML-Academics/ | 1BM18CS148_PSaiDee | Inbox (3,068) - deeksh | Inbox (242) - saidekshi | Writeup - Google Driv | B4 batch ML_LAB TES | Meet - vxc-qqvd-i

localhost:8888/notebooks/ML-Academics/1BM18CS148_PSaiDeekshith_ML_Lab_CIE2.ipynb#Training

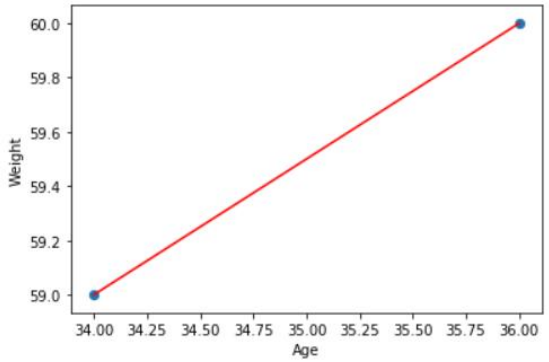
jupyter 1BM18CS148_PSaiDeekshith_ML_Lab_CIE2 Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

In [65]:
```
#Visualization
reg2 = LinearRegression()
reg2.fit(X_test[['Age']], y_test.Weight)
plt.scatter(X_test.Age,y_test.Weight,s=40)
plt.plot(X_test.Age, reg2.predict(X_test[['Age']]), color='red')
plt.xlabel('Age')
plt.ylabel('Weight')
```

Out[65]: Text(0, 0.5, 'Weight')



In [66]:
```
#Visualization
reg3 = LinearRegression()
reg3.fit(X_test[['Height']], y_test.Weight)
plt.scatter(X_test.Height,y_test.Weight,s=40)
plt.plot(X_test.Height, reg3.predict(X_test[['Height']]), color='red')
plt.xlabel('Height')
plt.ylabel('Weight')
```

Out[66]: Text(0, 0.5, 'Weight')

Prediction

Question : Height = 5.5, Age = 38, Weight = ?

In [63]:
```
model.predict([[38, 5.5]])
```
Out[63]:
```
array([[69.39177868]])
```

```
Math behind it :
```

In [67]:
```
#y=m1*x1+m2*x2+b
```

Slopes:

In [68]:
```
model.coef_
```
Out[68]:
```
array([[1.55923278, 4.66746509]])
```

Intercept:

In [69]:
```
model.intercept_
```
Out[69]:
```
array([-15.53012489])
```

In [76]:
```
y = model.coef_[:,0]*38 + model.coef_[:,1]*5.5 + model.intercept_
y
```
Out[76]:
```
array([69.39177868])
```

Accuracy

In [49]:
```
model.score(X_test, model.predict(X_test))
```
Out[49]:
```
1.0
```