DSBDA Practical4 Synthesis

May 12, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
#Create a Dataframe with Dependent Variable(x) and independent variable y.
x=np.array([95,85,80,70,60])
y=np.array([85,95,70,65,70])
```

[4]:
```
#Create Linear Regression Model using Polyfit Function:
model= np.polyfit(x, y, 1)
```

[5]:
```
#Observe the coefficients of the model.
model
```

[5]:
```
array([ 0.64383562, 26.78082192])
```

[6]:
```
#Predict the Y value for X and observe the output.
predict = np.poly1d(model)
predict(65)
```

[6]:
```
68.63013698630135
```

[7]:
```
#Predict the y_pred for all values of x.
y_pred= predict(x)
y_pred
```

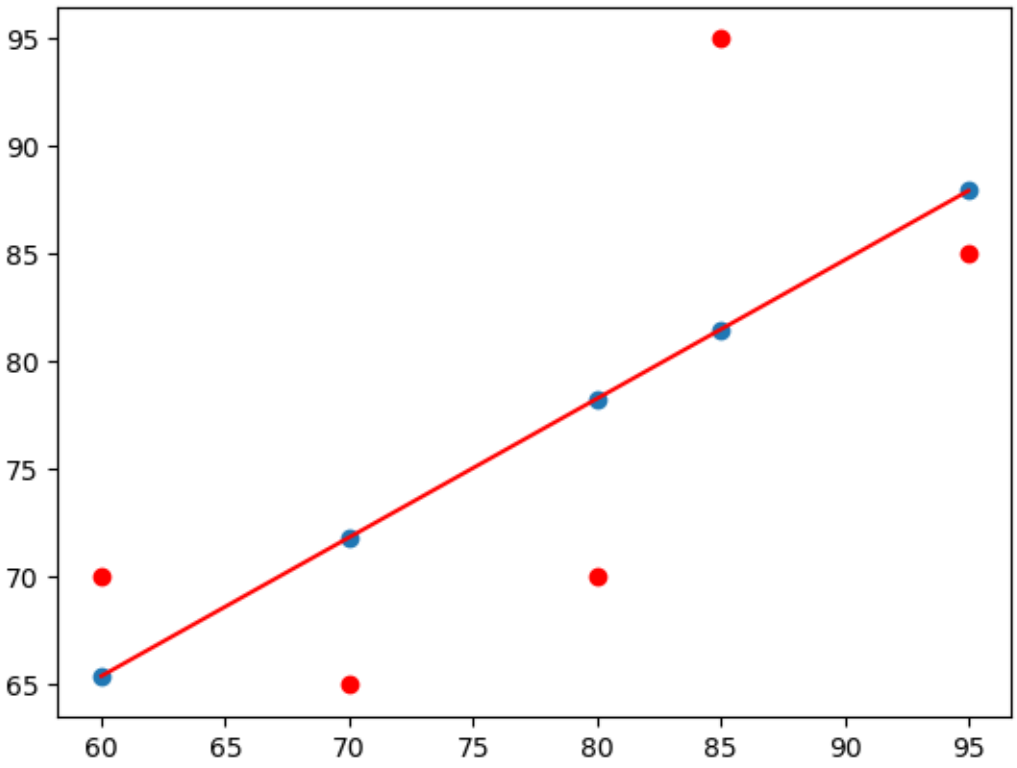
[7]:
```
array([87.94520548, 81.50684932, 78.28767123, 71.84931507, 65.4109589 ])
```

[10]:
```
#Evaluate the performance of Model (R-Suare)
#R squared calculation is not implemented in numpy... so that one should be␣
 ↪borrowed
 #from sklearn.
from sklearn.metrics import r2_score
r2_score(y, y_pred)
```

[10]:
```
0.4803218090889323
```

[11]:
```
#Plotting the linear regression model
y_line = model[1] + model[0]* x
plt.plot(x, y_line, c = 'r')
plt.scatter(x, y_pred)
plt.scatter(x,y,c='r')
```

[11]: <matplotlib.collections.PathCollection at 0x1cb630d7c40>



[ ]: