Name: Tarun Tanmay Roll no: N049 MBA Tech CE 3rd Year

#EXP 1

# importing packages import numpy as np

from sklearn.linear\_model import LinearRegression import matplotlib.pyplot as plt

# independent var x

x=np.array([5,15,25,35,45,55])

# retaining original value of x xm=x

#x.reshape()

#reshape x to any no of rows but 1 col x=x.reshape((-1,1))

#x.shape #print(x)

# dependent variable y

y=np.array([11,16,18,30,22,38])

# retaining original value of y ym=y

plt.scatter(x,y) plt.plot(x,y,"b.") #plt.show()

#plt.xlabel("Independent Variable, x-->") #plt.ylabel("Dependent Variable, y-->")

#plt.title("Data Points")

# Creating a linear regression model model=LinearRegression()

# fit model using given independent and dependent variables model.fit(x,y)

print("Intercept of the line: ",model.intercept\_) #y-intercept of the line,b0 print("Slope of the line: ",model.coef\_) #slope of the line,b1

#determining coeff of determination, r^2 r\_sq=model.score(x,y)

print("Coefficent of determination: ",r\_sq)

# predicting output for the given input x\_predict=x

y\_predict=model.predict(x\_predict)

#print actual output

print("Actual output",y,sep='\n')

#print predicted output

print("Predicted output",y\_predict,sep='\n')

plt.scatter(x,y) plt.plot(x,y,"b.")

plt.plot(x\_predict,y\_predict,"r-")

plt.xlabel("Independent Variable, x-->") plt.ylabel("Dependent Variable, y-->") plt.title("Data Points")

 Intercept of the line: 8.357142857142856 Slope of the line: [0.47142857]

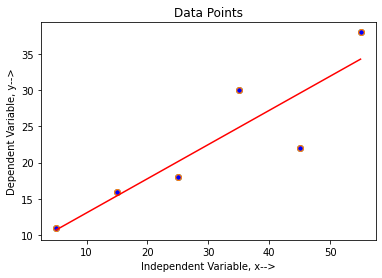
Coefficent of determination: 0.7913094027030955 Actual output

[11 16 18 30 22 38]

Predicted output

[10.71428571 15.42857143 20.14285714 24.85714286 29.57142857 34.28571429]

Text(0.5, 1.0, 'Data Points')



# computing b0 and b1 using formula

xmean=np.average(xm) ymean=np.average(ym)

xymean=np.average(np.multiply(xm,ym)) #print(xmean,ymean,xymean)

xmeansquare=xmean\*xmean

xsquarebar=np.average(np.multiply(xm,xm)) print(xmeansquare,xsquarebar)

# calculating b1

b1=((xmean\*ymean)-xymean)/(xmeansquare-xsquarebar) print("Slope of the line: ",b1)

# calculating b0

b0=ymean-(b1\*xmean)

print("Intercept of the line: ",b0)

plt.scatter(x,y) plt.plot(x,y,"b.")

plt.plot(x\_predict,y\_predict,"r-")

plt.xlabel("Independent Variable, x-->") plt.ylabel("Dependent Variable, y-->") plt.title("Data Points")

# plotting ybar line

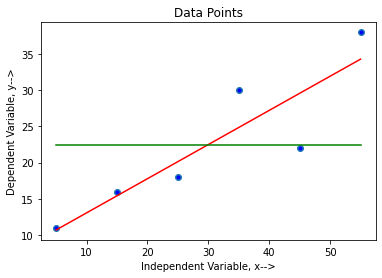
ybar=np.array([ymean,ymean,ymean,ymean,ymean,ymean]) plt.plot(x,ybar,"g-")

 900.0 1191.6666666666667

Slope of the line: 0.4714285714285713

Intercept of the line: 8.357142857142861

[<matplotlib.lines.Line2D at 0x7f3f171c9cf8>]

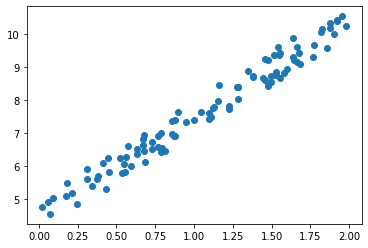


x=2\*np.random.rand(100,1)

y=4+(3\*x)+np.random.rand(100,1) plt.scatter(x,y)



<matplotlib.collections.PathCollection at 0x7f3f16be5390>



Conclusion :

1. The given data points have positive association.
2. Coeficients of the linear regression line can be determinte using linear regression model of sqlearn library. These coeficients are same as that of the one's using classical method.
3. Since coeficient of determination that is r square is 79.13 for the given data which shows

that linear regression can eliminate most of the error, however, the relation between x and y is not perfect.For a perfect relation value of r square should be 1.

1. R2 value cannot be improved by linear regression only.