1. **Linear Regression**

slope, intercept, r, p, std\_err = stats.linregress(x, y)

def myfunc(x):  
  return slope \* x + intercept  
  
mymodel = list(map(myfunc, x))

plt.scatter(x, y)

plt.plot(x, mymodel)

print(r);

1. **Polynomial Regression**

mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))  
  
myline = numpy.linspace(1, 22, 100)

plt.scatter(x, y)  
plt.plot(myline, mymodel(myline))

print(r2\_score(y, mymodel(x)))

1. **Multiple Regression**

df = pandas.read\_csv("cars.csv")

X = df[['Weight', 'Volume']]  
y = df['CO2']

regr = linear\_model.LinearRegression()  
regr.fit(X, y)

predictedCO2 = regr.predict([[2300, 1300]])

print(regr.coef\_)

1. **Clustering**

StudentData = pd.read\_csv('Student.csv')

data\_scaled = normalize(StudentData)

data\_scaled = pd.DataFrame(data\_scaled, columns=StudentData.columns)

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

#here we made the Dendrograms

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

dend = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

#here we defined the Threshold Line

plt.axhline(y=0.115, color='r', linestyle='--')

cluster = AgglomerativeClustering(n\_clusters=5, affinity='euclidean', linkage='ward')

cluster.fit\_predict(data)