**Practical No: 1**

**Design the Machine Learning Model**

**AIM: Design a simple machine learning model to train the training instances and test the same.**

**Description:**

1. Training Data

Training data is the data you use to train an algorithm or machine learning model to predict the outcome you design your model to predict.

Training data is always more or equal in size than test data

1. Test Data

Testing data is used to evaluate our model performance.

**Code with output**

import numpy

import matplotlib.pyplot as plt

numpy.random.seed(2)

x = numpy.random.normal(3,1,100)

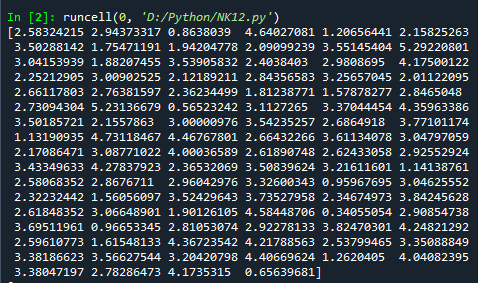
print(x)

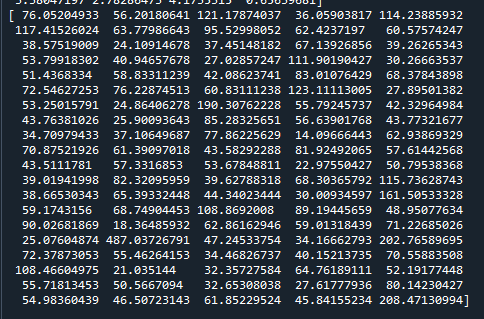
y = numpy.random.normal(150,40,100) /x

print(y)

plt.scatter(x,y)

plt.show()





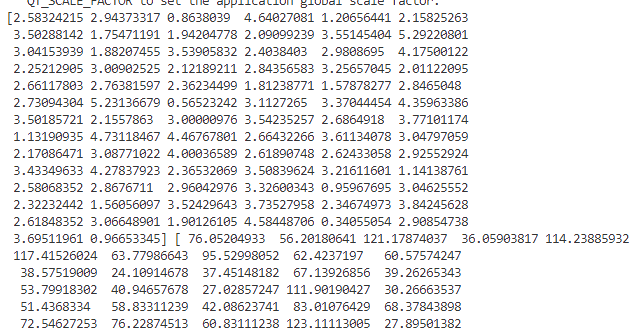
train\_x = x[:80]

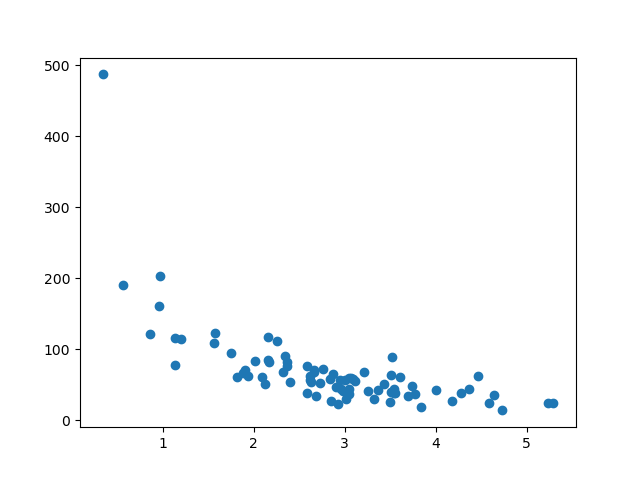
train\_y = y[:80]

test\_x = x[:20]

test\_y = y[:20]

print(train\_x,train\_y,test\_x,test\_y)





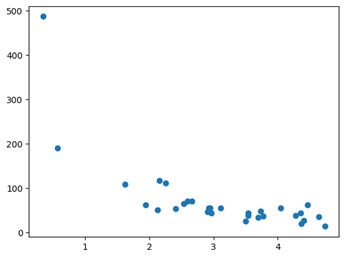
plt.scatter(train\_x,train\_y)

plt.show()

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,test\_size=0.3)

plt.scatter(test\_x,test\_y)

plt.show()



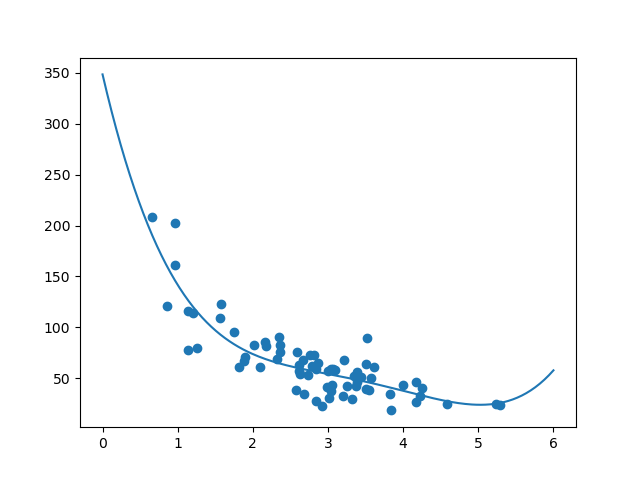
mymodel = numpy.poly1d(numpy.polyfit(train\_x, train\_y, 4))

myline = numpy.linspace(0,6,200)

plt.scatter(train\_x, train\_y)

plt.plot(myline, mymodel(myline))

plt.show()



mymodel = numpy.poly1d(numpy.polyfit(test\_x, test\_y, 4))

myline = numpy.linspace(0,6,200)

plt.scatter(test\_x, test\_y)

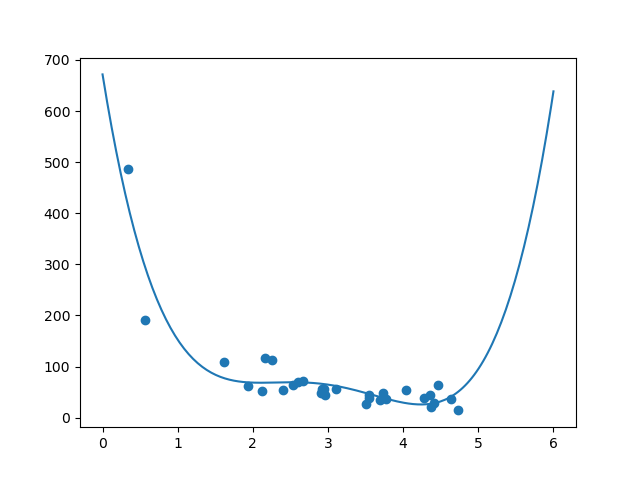
plt.plot(myline, mymodel(myline))

plt.show()

r2 = r2\_score(train\_y, mymodel(train\_x))

print(r2)

print(mymodel(5))





# As we get high rscore the model is working good

Learnings

First we have created random data for x and y.

Then we have divided it into train test part with 80:20 ratio. visualizes the data and the fitted models.

Then after fitting model, we have evaluated model performance using r square. Then make prediction using trained model