**Polynomial Regression**

Things to learn

1. numpy.linspace(1,101,11)
2. creates a numpy array = nd array starting from 1 to 101 with 11 points, 10 intervals 2. numpy.polyfit(x,y,2)

Fits a quadratic curve to the given x,y data and returns a ax^2 + bx + c type array

1. numpy.poly1d returns an equation from the polyfit nd array
2. Find correlation coefficient

from sklearn.metrics

import r2\_score correlation=r2\_score(y, model(x))

1. Matplotlib

Make many plots in the same figure

plt.plot(x,y,"b-",label="Original")

x,y,color,label

plt.scatter(x,y,color="yellow")

Program

import numpy

import matplotlib.pyplot as plt

from sklearn.metrics

import r2\_score x = [1,2,3,4]

y = [1,4,9,14]

print(numpy.linspace(1,101,11))

model = numpy.poly1d(numpy.polyfit(x, y, 2))

print(model)

print(numpy.polyfit(x, y, 1))

plotx =x#[1,2,3,4,5,6,7]

# numpy.linspace(1, 22, 100)

correlation=r2\_score(y, model(x))

print("Correlation",correlation)

plt.scatter(x,y,color="yellow")

plt.plot(x,y,"b-",label="Original")

plt.plot(plotx, model(plotx),"r-",label="Predicted")

plt.ylabel('Y')

plt.xlabel('X')

plt.legend()

plt.show()

xvalue=4

predict=model(xvalue)

print(predict)