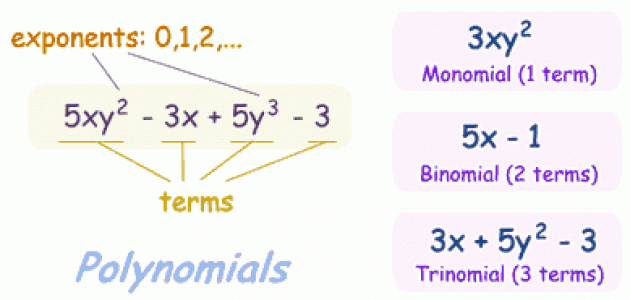
Polynomial Regression

Short note about Polynomial Regression:

A polynomial is an expression consisting of variables (also called indeterminates) and coefficients, that involves only the operations of addition, subtraction, multiplication, and non-negative integer exponents of variables. An example of a polynomial of a single indeterminate x is x2 − 4x + 7.

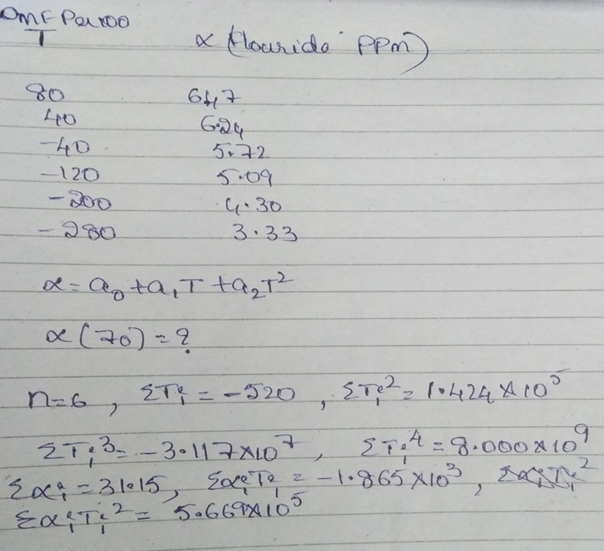


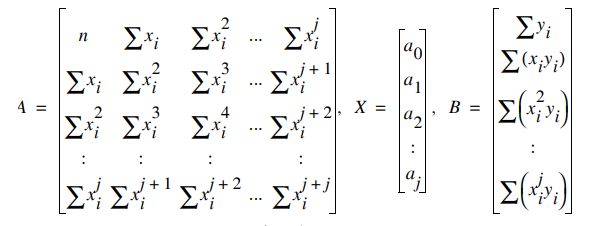
*Polynomial regression Formula (y = β0 + β1x + β2)*

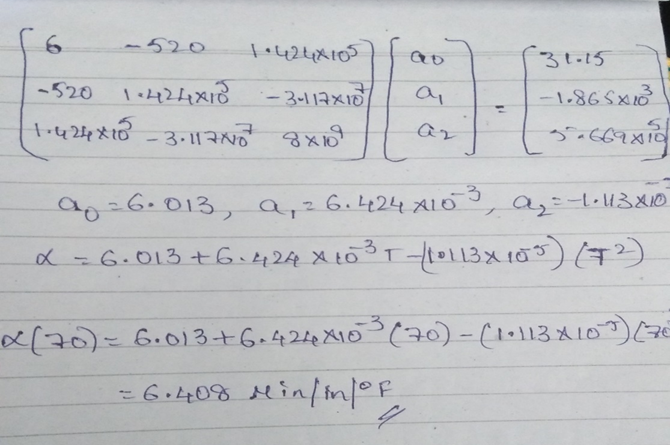
(ii) IN Calculation

|  |  |
| --- | --- |
| Flourideppm (X) | DMF per 100 (Y) |
| 80 | 6.47 |
| 40 | 6.24 |
| -40 | 5.72 |
| -120 | 5.09 |
| -200 | 4.3 |
| -280 | 3.33 |

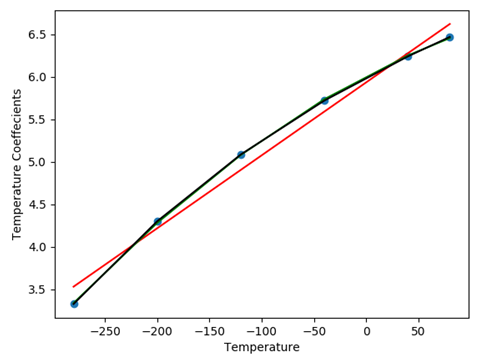
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **DMF per 100(x)** | **Flourideppm(y)** | **x^2** | **x^3** | **x^4** | **xy** | **y^2** | **xy^2** |
| 80 | 6.47 | 6400 | 512000 | 40960000 | 517.6 | 41.8609 | 3348.872 |
| 40 | 6.24 | 1600 | 64000 | 2560000 | 249.6 | 38.9376 | 1557.504 |
| -40 | 5.72 | 1600 | -64000 | 2560000 | -228.8 | 32.7184 | -1308.736 |
| -120 | 5.09 | 14400 | -1728000 | 207360000 | -610.8 | 25.9081 | -3108.972 |
| -200 | 4.3 | 40000 | -8000000 | 1600000000 | -860 | 18.49 | -3698 |
| -280 | 3.33 | 78400 | -21952000 | 6146560000 | -932.4 | 11.0889 | -3104.892 |
| -520 | 31.15 | 142400 | -31168000 | 8000000000 | -1864.8 | 169.0039 | -6314.224 |







Graph



**USE CASES OF Polynomial Regression**

To find Truth or Bluff about salary with experience

To find Yield using Temperature

To find prize using Size

To find Pizza diameter using pizza price

PYTHON CODE WITHOUT LIBRARY

Polynomial Regression

import numpy as np

import scipy as sc

import pandas as pd

import matplotlib.pyplot as plt

data=pd.read\_csv("poly-test.csv",sep='\t')

X=np.array(data['X'].values)

Y=np.array(data['Y'].values)

print(X)

print(Y)

# we can use np or sc as both numpy and scipy have most or all of its methods and functions same

linear = sc.polyfit(X,Y,1)

linear\_reg=sc.poly1d(linear) # in order to apply the coeffecients to X to get a proper equation

print(linear)

print(linear\_reg)

print("\n")

#for degree 2

quadratic =sc.polyfit(X,Y,2) # fitting the model

quadratic\_reg=sc.poly1d(quadratic) # in order to apply the coeffecients to X to get a proper equation

print(quadratic)

print(quadratic\_reg)

print("\n")

#for degree 3

cubic =sc.polyfit(X,Y,3)

cubic\_reg=sc.poly1d(cubic) # in order to apply the coeffecients to X to get a proper equation

print(cubic)

print(cubic\_reg)

print("\n")

plt.plot(X,Y,'o')

plt.plot(X,linear\_reg(X),color='red')

plt.plot(X,quadratic\_reg(X),color='green')

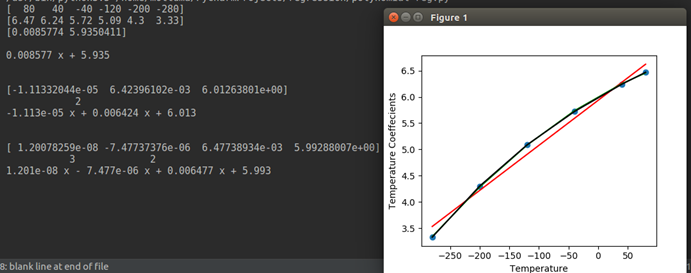
plt.plot(X,cubic\_reg(X),color='black')

plt.xlabel('Temperature')

plt.ylabel('Temperature Coeffecients')

plt.show()

Visualization



**Python Code With library**

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(x, y)

from sklearn.preprocessing import PolynomialFeatures

poly\_reg = PolynomialFeatures(degree = 4)

X\_poly = poly\_reg.fit\_transform(x)

poly\_reg.fit(X\_poly, y)

lin\_reg = LinearRegression()

lin\_reg.fit(X\_poly, y)

plt.scatter(x, y, color = 'red')

plt.plot(x, lin\_reg.predict(poly\_reg.fit\_transform(x)), color = 'blue')

plt.title('salary')

plt.xlabel('Exp')

plt.ylabel('salary')

plt.show()

lin\_reg.predict(poly\_reg.fit\_transform(6.5))

