**Intelligent Systems Lab**

**Lab Exam**

Name- Adhyyan Tripathi

Roll no -8

Sec C

Reg no – 201700403

**Algorithm:**

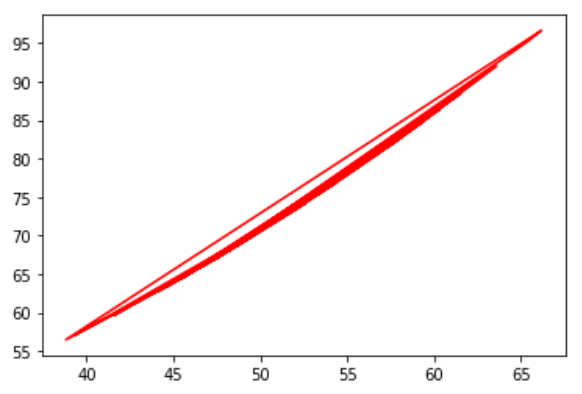
Steps involved in polynomial curve fitting are –

* Start
* Data Pre-processing: removing redundant data, removing skewness, removing null values.
* Transform the features into polynomial features with specified degree.
* Build a linear Regression model and fit it to the dataset
* Apply the Ridge Regularization with appropriate value of lambda.
* Again Calculate the value of RMSE and R2.
* Visualize the result for Polynomial Regression model.
* Stop.

Q.1) Write a program to implement the concepts of regression via polynomial curve fitting.

Ans- Applying polynomial curve fitting for degree 2,5 and 10 on 30 datapoints we get –

**For degree 2** **without regularization and with regularization –**



RMSE for without regularization = 10.706534281222241

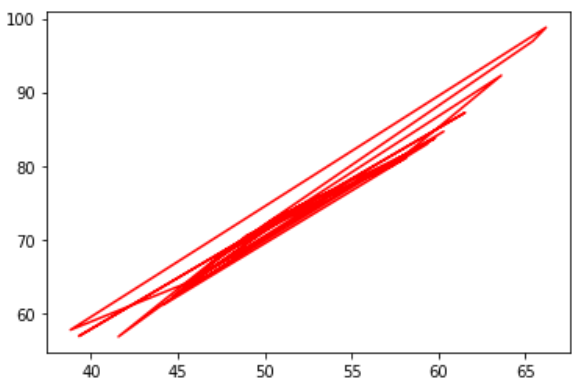
R2 for without regularization = 0.5076525817516899

RMSE for with regularization = 10.706534281222247

R2 for with regularization = 0.5076525817516893

The value of R2 and RMSE shows that it is an underfit.

**For degree 5 without regularization and with regularization –**



RMSE for without regularization = 10.624525208966704

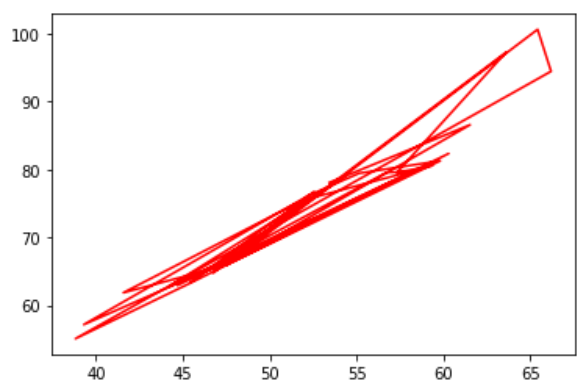
R2 for without regularization = 0.5151661834128077

RMSE for with regularization = 10.878469180116237

R2 for with regularization = 0.4917125199330582

The value of R2 and RMSE shows that it is an underfit.

**For degree 10 without regularization and with regularization –**



RMSE for without regularization = 10.413746693104637

R2 for without regularization = 0.5342124670276756

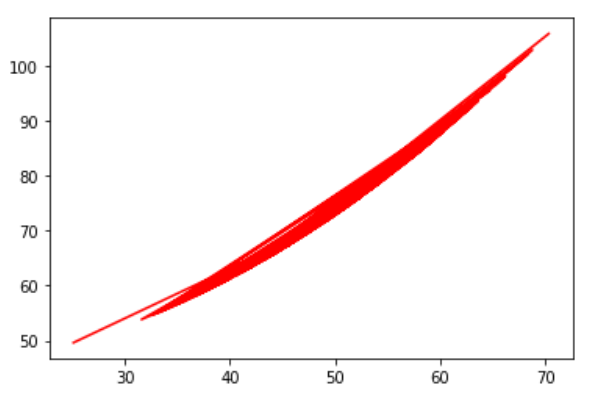
RMSE for with regularization = 10.885721921914993

R2 for with regularization = 0.4917125199330582

The value of R2 and RMSE shows that it is an underfit.

Applying polynomial curve fitting for degree 2,5 and 10 on 100 datapoints we get –

**For degree 2 without regularization and with regularization–**



RMSE for without regularization = 10.23104645312165

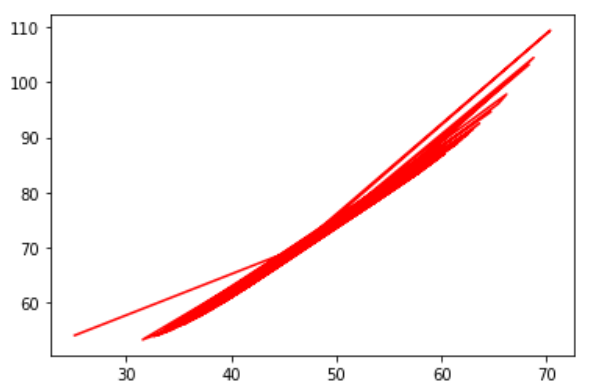
R2 for without regularization = 0.5979020408762089

RMSE for with regularization = 11.083581322946797

R2 for with regularization = 0.5280978238956777

The value of R2 and RMSE shows that it is an underfit.

**For degree 5 without regularization and with regularization–**



RMSE for without regularization = 10.194055364104672

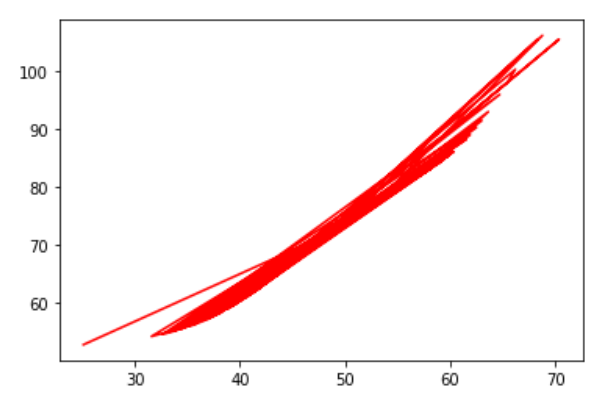
R2 for without regularization = 0.6008044130589294

RMSE for with regularization = 10.485510222023937

R2 for with regularization = 0.5776515626645728

The value of R2 and RMSE shows that it is an underfit.

**For degree 10 without regularization and with regularization–**



RMSE for without regularization = 10.164219180954397

R2 for without regularization = 0.6031377421030951

RMSE for with regularization = 10.542707277783544

R2 for with regularization = 0.5730312869799582

The value of R2 and RMSE shows that it is a bestfit.

**Inference-**

The data above shows that for the given dataset the model is underfitted for degree 2 and 5 and it is best fitted for degree 10.

When applied L2 ridge regularization with non-zero value of lambda the R2 value is generally observed lower than the non-regularized R2.

The best value of lambda is observed to be 0.89.

**Code –**

from sklearn.linear\_model import Ridge

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import random

import operator

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import PolynomialFeatures

dataset = pd.read\_csv('Dataset\_Regression.csv')

x = dataset.iloc[0:100, 0:1].values

y = dataset.iloc[0:100, 1:2].values

x.reshape(-1, 1)

polynomial\_features= PolynomialFeatures(degree=50)

x\_poly = polynomial\_features.fit\_transform(x)

lm = LinearRegression()

lm.fit(x\_poly, y)

y\_poly\_pred = lm.predict(x\_poly)

plt.plot(x,y\_poly\_pred,color='red')

rmse = np.sqrt(mean\_squared\_error(y,y\_poly\_pred))

r2 = r2\_score(y,y\_poly\_pred)

print("RMSE and R2 without Regularization-")

print(rmse)

print(r2)

ridgereg = Ridge(alpha=1, normalize=True)

ridgereg.fit(x\_poly, y)

y\_pred = ridgereg.predict(x\_poly)

rmse = np.sqrt(mean\_squared\_error(y,y\_pred))

r2 = r2\_score(y,y\_pred)

print("RMSE and R2 with Regularization-")

print(rmse)

print(r2)