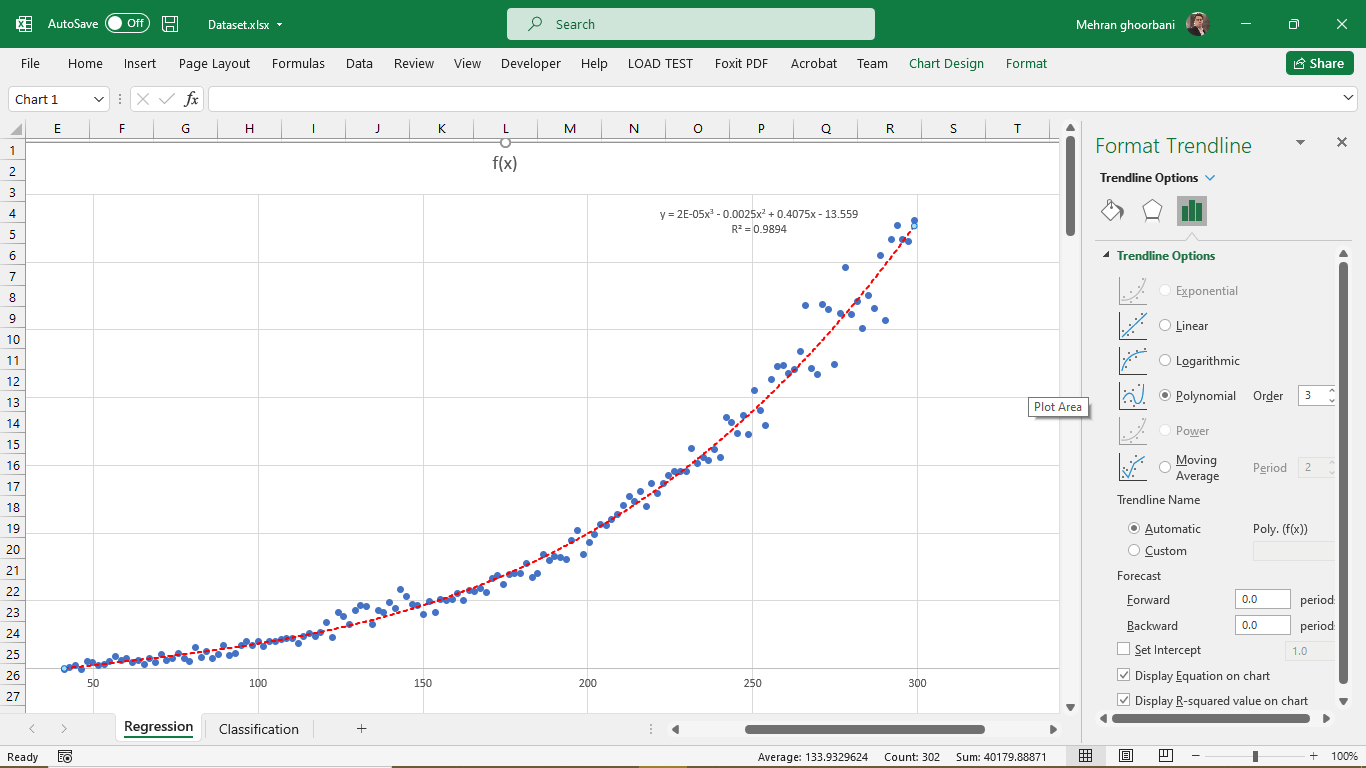
**آنالیز اولیه داده ها**

**Excel**

ابتدا برای بررسی دیتای مورد نظر را درون اکسل بررسی میکنیم تا یک دید کلی از روند انجام کار را داشته باشیم.



با توجه به معیار های داده شده از اکسل میتوان نتیجه گرفت، معادله درجه 3 مناسب برای برازش (regression) میباشد که r-squared مناسبی را برای تخمین ما در نظر گرفته است.

**MATLAB**

%% Clear memory

% This problem is solved by Mehran Ghorbani for Soft Computing lesson

% in school of civil engineering in iran university of science and

% technology (IUST)

clear;

clc;

close all;

%% Load/Plot Data

data = load('dataset.txt');

msk = rand(size(data,1),1) < 0.8; % Split data

%% Train Set

data\_train = data .\* msk;

x\_train = data\_train(:,1);

x\_train(x\_train == 0)= [];

y\_train = data\_train(:,2);

y\_train(y\_train == 0)= [];

x\_train = x\_train./max(x\_train); % Normalize x data

y\_train = y\_train./max(y\_train); % Normalize y data

x\_min\_train = min(x\_train); % minimum x\_train data

y\_min\_train = min(y\_train); % minimum y\_train data

%% The Developement Set

data\_test = data .\* ~msk;

x\_test = data\_test(:,1);

x\_test(x\_test == 0)= [];

y\_test = data\_test(:,2);

y\_test(y\_test == 0)= [];

x\_test = x\_test./max(x\_test); % Normalize x data

y\_test = y\_test./max(y\_test); % Normalize y data

x\_min\_test = min(x\_test); % minimum x\_test data

y\_min\_test = min(y\_test); % minimum y\_test data

%% Obtain Regression Coefficients and R^2

[f\_train,gof,output] = fit(x\_train,y\_train,'poly3'); % Cubic regression is useful for this excercise

f = string(formula(f\_train)); % get the formular in string format

names = coeffnames(f\_train); % get the coeff names

vals = coeffvalues(f\_train); % get those values

for i = 1:length(names)

f = strrep(f,string(names(i)),string(vals(i))); % replace the coeff names with values

end

f

y\_hat = f\_train(x\_test);

figure

P1 = plot(x\_train,y\_train,'b.',...

'MarkerSize',15)

grid on;

hold on

P2 = plot(x\_test,y\_test,'r.',...

'MarkerSize',15)

P3 = plot(f\_train,'m','predobs')

xlabel('x (ND)','fontsize',14,'Interpreter','latex')

ylabel('y (ND)','fontsize',14,'Interpreter','latex')

title('Polynomial Regression','fontsize',14,'Interpreter','latex')

legend('Train Set','Test Set','Fitted Polynomial','Prediction Bound','location','northwest','fontsize',14,'Interpreter','latex')

set(gca,'TickLabelInterpreter','latex')

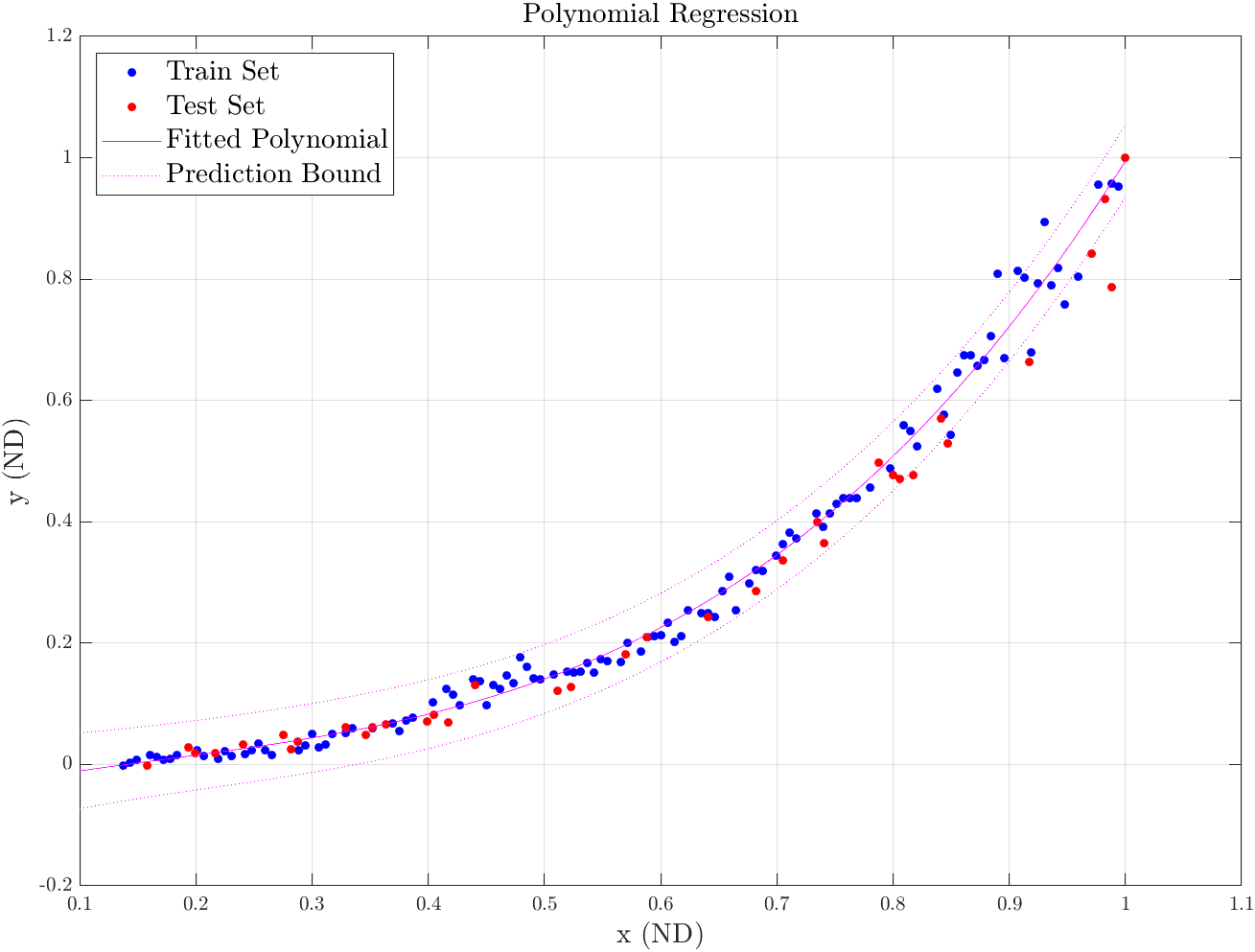
set(gcf,'Position',[0 0 1000 1000]) % Will probably need to change this on your computer

% Extract coefficients

error = abs(y\_hat-y\_test).^2

coefs = coeffvalues(f\_train); % Extract curve fit coefficients

Results = gof % Display the R^2 value. It's very close to 1, which is "good"



***f =***

"1.2744\*x^3 + -0.58617\*x^2 + 0.33101\*x + -0.037141"

***error =***

0.0000

0.0000

0.0000

0.0000

0.0001

0.0000

0.0006

0.0000

0.0000

0.0005

0.0000

0.0041

0.0000

0.0003

0.0000

0.0004

0.0005

0.0000

0.0000

0.0016

0.0003

0.0001

0.0005

0.0000

0.0000

0.0001

0.0001

***Results =***

struct with fields:

sse: 0.1173

rsquare: 0.9883

dfe: 119

adjrsquare: 0.9880

rmse: 0.0314

**PYTHON (JUPYTER NOTEBOOK)**