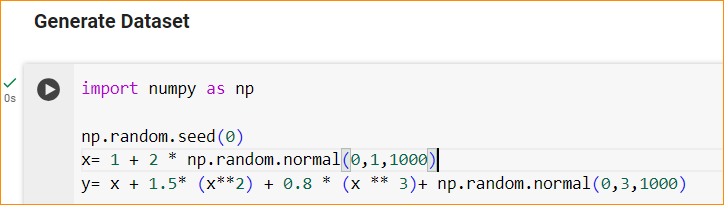
**Polynomial Regression – Python**

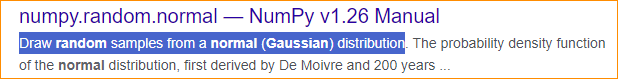
In here we will implement polynomial regression model and check the data and choose the right model.

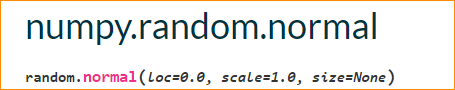
1. Generate a Random data

you've created two arrays, x and y, with x being drawn from a normal distribution and y being calculated based on a polynomial relationship with some added noise.



Using numpy random dataset, use noramal()- in here generate data in the Gaussian distribution.





Here – 0 is mean

1. Stand deviation

1000 – how many number of data we care

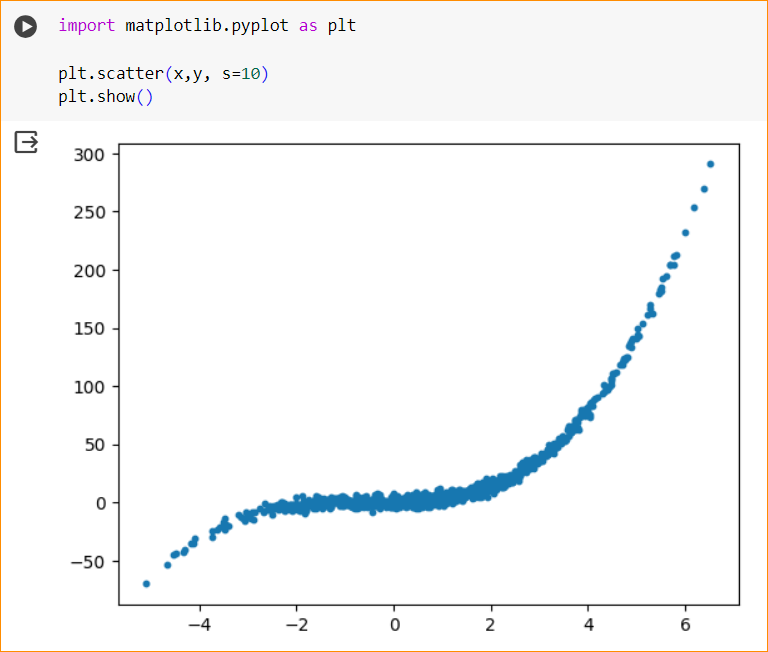
After X define

I have define pre determine a function in the order of 3

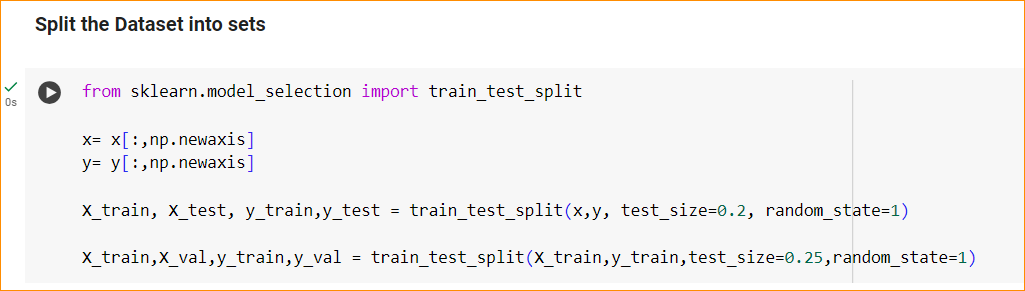


Using this I have generate random data, fit to a polynomial function of degree

1. Plot the data and check how it look like



1. Split the data in to 3 potions



SikitLearn does not have function to split data into 3 potions. So now we divide data in to 3 potions.

Training and testing

Here use 20 % allocated to testing and 80% for training.

* Now training data divide in to 2 – for Training and validation



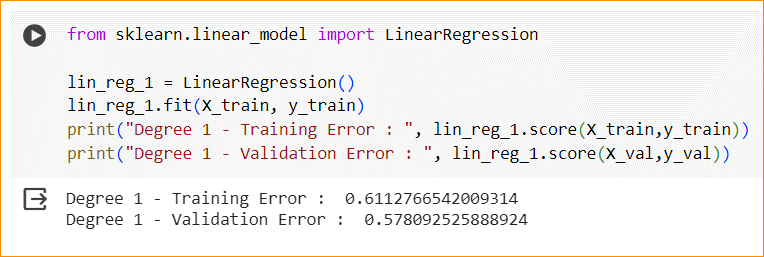
So 60% for training and 20 for validation

80 / (25/100) = 80/4 = 20 %

80-20 = 60%

1. 1st go with degree one Regression

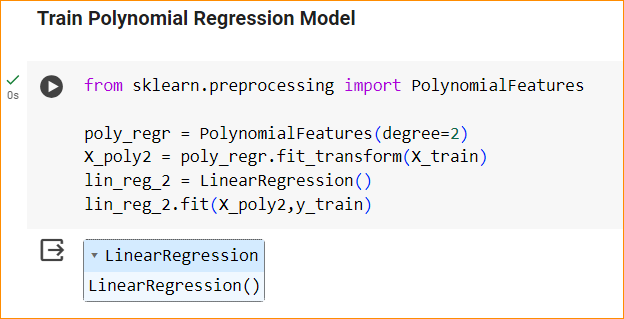
Degree one mean simple linear regression

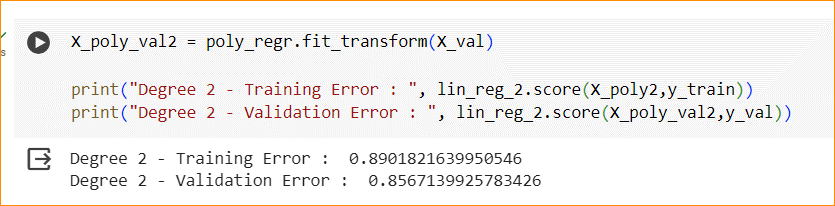


In here the training error not close to 1 also validation error-

That mean degree one is not suitable and it not performing well for this data.

1. Increased one degree (2nd degree) – Polynomial Regression





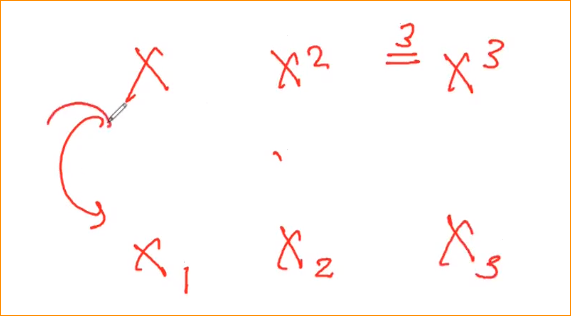
In here ,

So we have to prepare features.

In here have special function call polynomial features, so have to specified the degree.

Once create a feature extractor, we can apply our data.

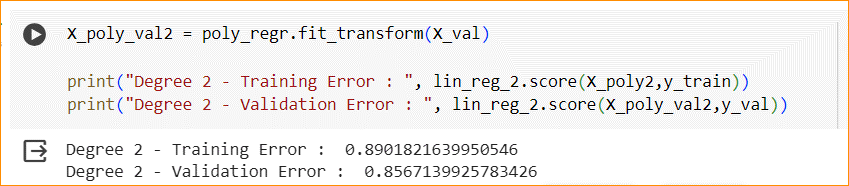
We want only transform X, no need to transform Y.



Then we can create a simple linear regression but with transform features (X= X1, X^2 = X2 ), and Fit that

And fit the model with X\_poly2 with Y\_train.

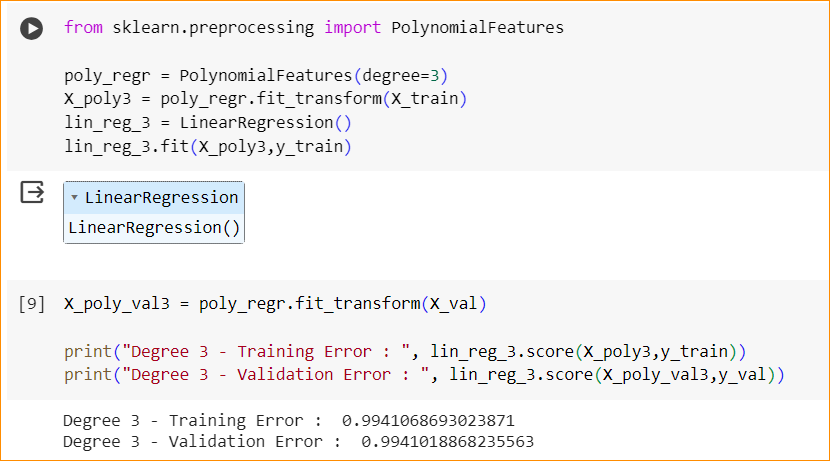
Again check errors –



So R^2 is significantly increase. Both

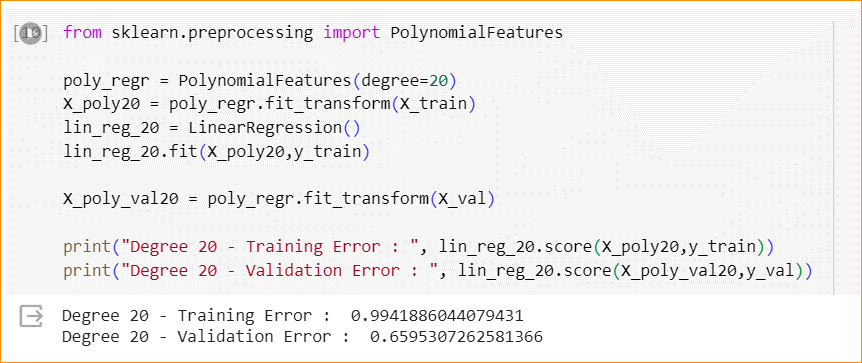
This mean your model is performing well.

1. Model degree increasing to 3



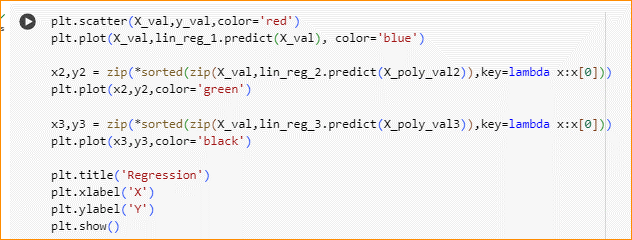
Can see still this is the best model.

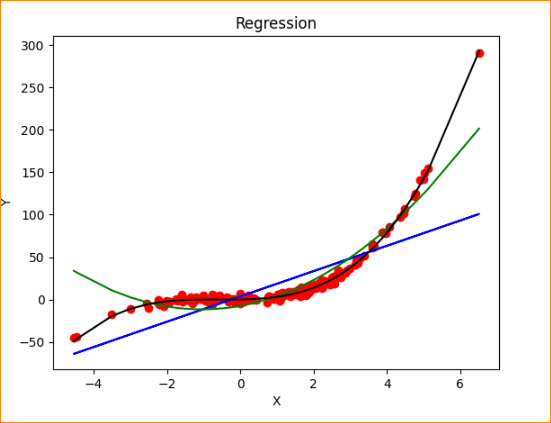
1. Model degree increasing to 20



In here both increasing but validation decrees because , overfitting our data.

1. Now plot the models





Blue – Degree 1(under fitting, not fitting the data)

Green = degree 2 (at the end it is not fitting to the data)

Black - degree 3 (good model)