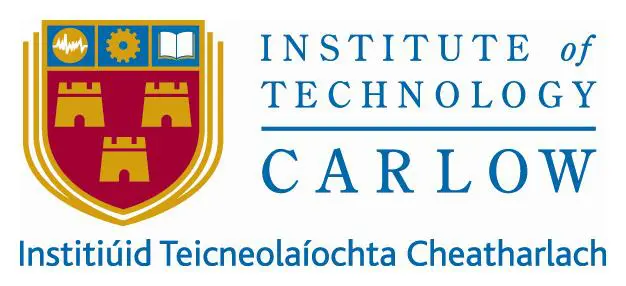
**Tutorial**



**Data Science Tutorial Project**

Name: Daniel Polak

Date: 26/03/2021

Supervisor: Greg Doyle

[**Introduction**](#_ab7t8iac1qf7) **1**

[**Polynomial Regression Tutorial**](#_wkc5y6kstge4) **3**

[Introduction](#_hfg8g0s78197) 3

[Resources](#_abwaaeq00568) 3

[Tutorial](#_rdwsktnknt3v) 3

[Conclusion](#_j4btyk5vepbd) 9

[**References**](#_ggc24o82fr11) **10**

# Introduction

This document contains four tutorials which detail how some machine learning algorithms work. These algorithms are Polynomial Regression, Decision Trees, Clustering and Neural Networks. Each tutorial consists of resources that are helpful in learning about each algorithm in more depth.

# **Polynomial Regression Tutorial**

## **Introduction**

In this tutorial I will show an example of Polynomial Regression using python and jupyter notebook.

**What is Polynomial Regression?** It is a form of regression analysis in which the relationship between variables x and y is modelled as nth degree polynomial. It fits the non-linear relationship between the value of x and y. While it fits non-linear model to the data, it is considered linear in the sense that the regression function E(*y* | *x*) is linear in the unknown parameters that are estimated from the data. Because of this, it is considered a special case of multiple linear regression [1].

**Why use polynomial regression?** It is best used when linear regression doesn’t capture patterns in data. This is called under-fitting.

In this tutorial, you will learn how to apply Polynomial Regression Algorithm to the data, process data. At the end I will show you how this algorithm can be changed into Linear Regression by changing two lines of code.

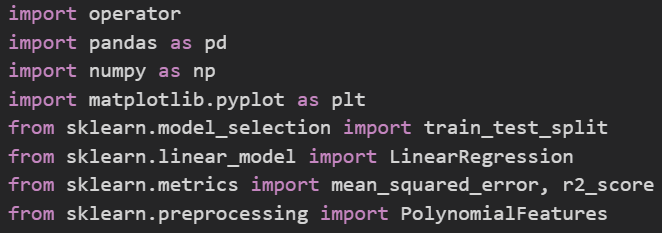
## **Resources**

Data for this tutorial: [Kaggle.com](https://www.kaggle.com/iqbalrony/polynomial-regression)

[Understanding Polynomial Regression!!!](https://medium.com/analytics-vidhya/understanding-polynomial-regression-5ac25b970e18)

## **Tutorial**

First we will need these import at the top of the file:



Pandas is used to read in our .csv file which contains data.

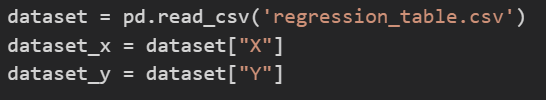
Numpy library will allow us to reshape our data.

Matplotlib is a library that will allow us to plot diagrams.

Sklearn library contains various Machine Learning algorithms which we will be using.

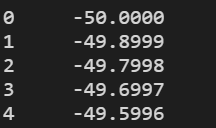
Except for the operator library, which is part of python, you will have to install all the libraries (Pandas, Numpy, Matplotlib, Sklearn) by using the pip command e.g. *pip install pandas*.

Next, we will read in our data from .csv file (link to the file) and split it into two arrays, each will contain one column from the file:

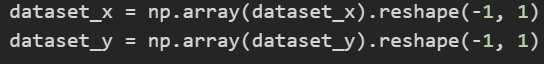


Be sure to put the .csv file into the same folder as your python file, otherwise the program will not be able to locate it. You can also change the path to the file instead.

Data will look like this:

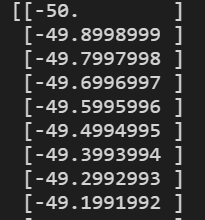


Now that we split the data into two columns, we will reshape them to form a 2D array:



We have to reshape our data because sklearn is not able to decide whether data that we passed is one row of data with multiple columns or multiple sets of data with one column. First, we will create an array from the columns by using np.array() method. Now we can reshape the array from 1D to 2D which will allow sklearn to use the data. We provide reshape with two arguments -1 and 1. -1 means that there is an unknown amount of rows and we want numpy to figure that amount out and 1 means that there is only one column.

Now data looks like this:

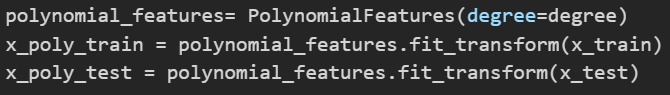


Now we can split each data set into training and testing sets:



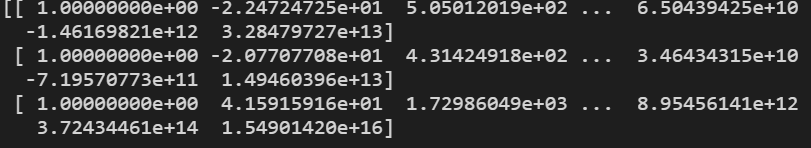
Test\_size means how much data we want to be put into testing sets. In the above you can see that in this case it is 0.33, this means that the split between training and testing sets is 67% and 33% or 670 and 330 data points.

Now we fit the data:



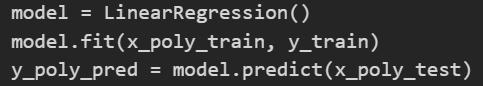
Here we supply degree n to polynomial features which best describes our data. We also fit and transform x\_train and x\_test.

It transforms data like so:



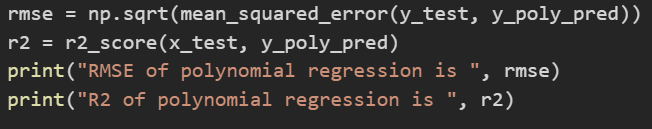
To learn more please visit <https://towardsdatascience.com/what-and-why-behind-fit-transform-vs-transform-in-scikit-learn-78f915cf96fe>

Here we use training sets and fit them into linear regression:

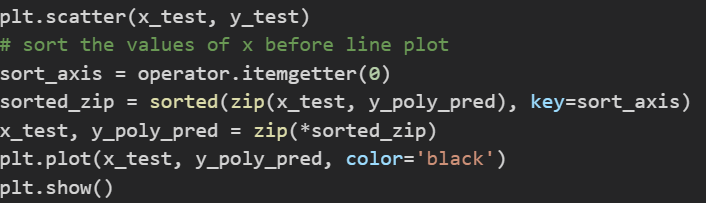


After we fit the training sets we use the x\_poly\_test to make a prediction.

Then we calculate R2(coefficient of determination) and RMSE (Root Mean Square Error):



Now we can finally plot the diagram and show the polynomial regression:



First we use matplotlib to scatter the test data on the diagram.

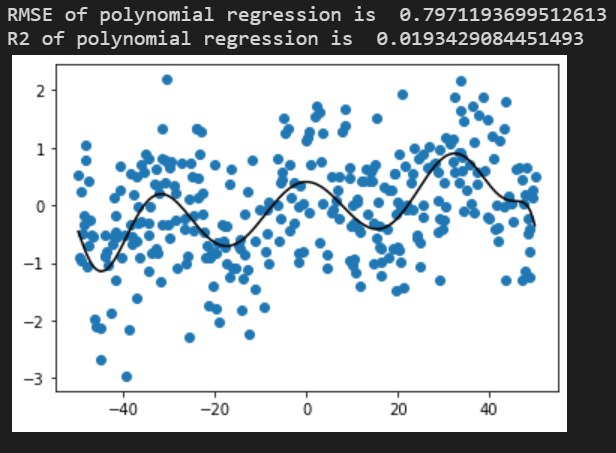
We then sort the data of x\_test and y\_poly\_pred.

Afterwards we plot the line and show the diagram.

We can run this in a jupyter notebook by typing polynomial\_regression(10).

This is due to us making it as a method in which you can replace the 10 for any number to see the effect of degree on the line.

Here is the final result:



Full Code:

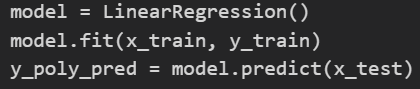


## Conclusion

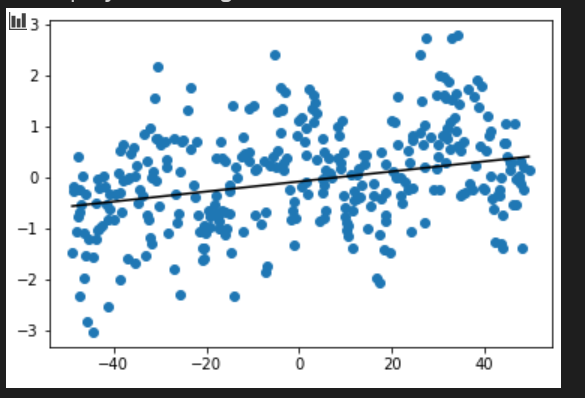
This concludes the tutorial to showcase polynomial regression. In this tutorial you hopefully learned how to read data from a file, separate the data, reshape the data and split the date into test and training sets. Also how to fit the data and predict the values by using training data.

To see different behavior of polynomial regression, you can use a dataset of different size.

By making a few changes to the code you can make this into linear regression.



Which will result in:



# References

[1] 25th December 2020, “Polynomial Regression”, “Wikipedia.com”, <https://en.wikipedia.org/wiki/Polynomial_regression>, (Accessed on 3rd March 2021)

