## **Introduction to Linear Regression in Python**

Linear regression is a type of supervised learning algorithm, commonly used for predictive analysis. As the name suggests, linear regression performs regression tasks. Now, what is regression? Well, regression is nothing but a technique that displays the relationship between two variables.

Here’s the table of contents for this module on Linear Regression in Python:

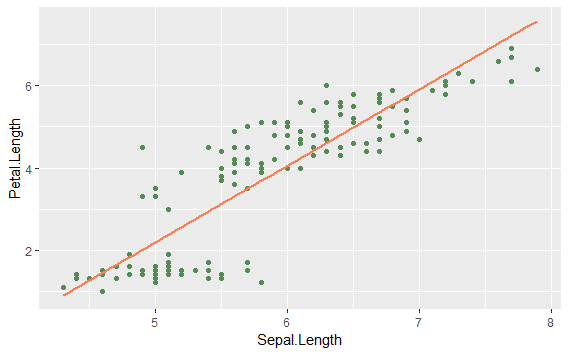
* **[What Is Linear Regression?](https://intellipaat.com/blog/what-is-linear-regression/" \l "What-Is-Linear-Regression)**
* **[Linear Regression Line of Best Fit](https://intellipaat.com/blog/what-is-linear-regression/" \l "Line-of-Best-Fit)**
* **[Regression CoefficientCoefficient of](https://intellipaat.com/blog/what-is-linear-regression/" \l "Coefficient-of-X)**
* **[Hands-on – Linear Regression Using Sklearn in Python – Boston Housing Data](https://intellipaat.com/blog/what-is-linear-regression/" \l "Boston-Housing-Prices-Dataset)**
  + **[Simple Linear Regression in Python](https://intellipaat.com/blog/what-is-linear-regression/" \l "Simple-Linear-Regression-Model)**
  + **[Multiple Linear Regression in Python](https://intellipaat.com/blog/what-is-linear-regression/" \l "Multiple-Linear-Regression-Model)**

## **What Is Linear Regression?**

As mentioned above, linear regression is a predictive modeling technique. It is used whenever there is a linear relation between the dependent and the independent variables.

Y = b0 + b1\* x

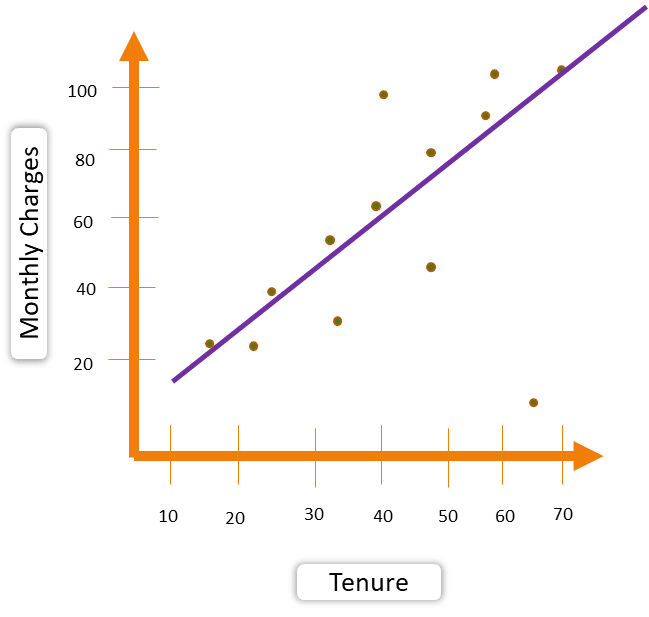
It is used in estimating exactly how much of y will change, when x changes a certain amount.



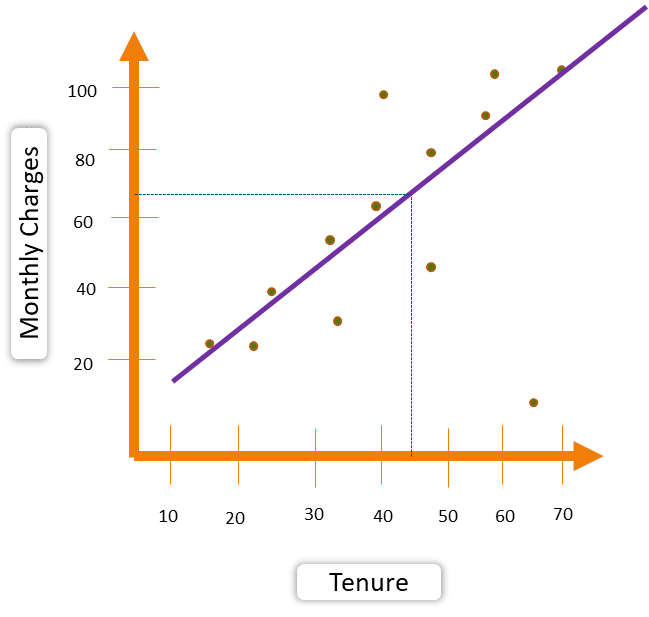
As we see in the picture, a flower’s sepal length is mapped onto the x-axis and the petal length is mapped on the y-axis. Let us try and understand how the petal length changes with respect to the sepal length with the help of linear regression. Let us have a better understanding of linear regression with another example given below.

### **Example:**

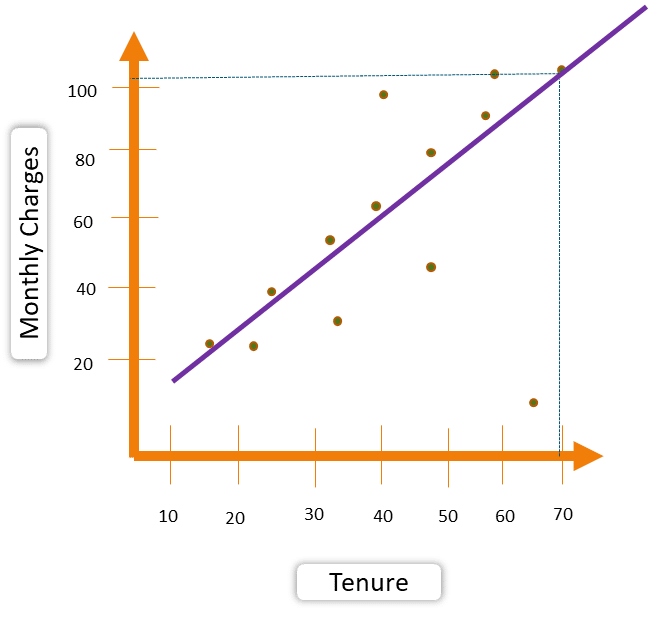
Say, there is a telecom network called Neo. Its delivery manager wants to find out if there’s a relationship between the monthly charges of a customer and the tenure of the customer. So, he collects all customer data and implements linear regression by taking monthly charges as the dependent variable and tenure as the independent variable. After implementing the algorithm, what he understands is that there is a relationship between the monthly charges and the tenure of a customer. As the tenure of the customer increases, the monthly charges also increase. Now, the best fit line helps the delivery manager find out more interesting insights from the data. With this, he can predict the values of y for every new value of x.



Let us say, the tenure of a customer is 45 months, and with the help of the best fit line the delivery manager can predict that the customer’s monthly charges would be somewhere around $64.



Similarly, if the tenure of a customer is 69 months, then with the help of the best fit line the delivery manager can predict that the customer’s monthly charges would be somewhere around $110.

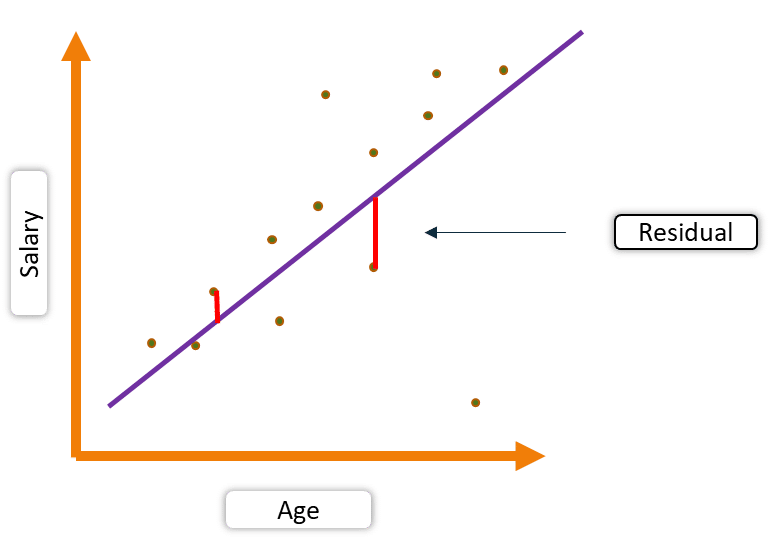


This is how linear regression works. Now, the question is how to find the best fit line?

## **Linear Regression Line of Best Fit**

The line of best fit is nothing but the line that best expresses the relationship between the data points. Let us see how to find the best fit line in linear regression.

This is where the residual concept comes into the picture which is shown in the image below:

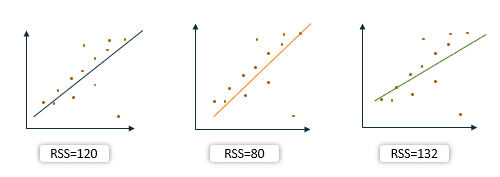


Red lines in the above image denote residual values, which are the differences between the actual values and the predicted values. How does residual help in finding the best fit line?

To find out the best fit line, we have something called **residual sum of squares (RSS)**. In RSS, we take the square of residuals and sum them up.



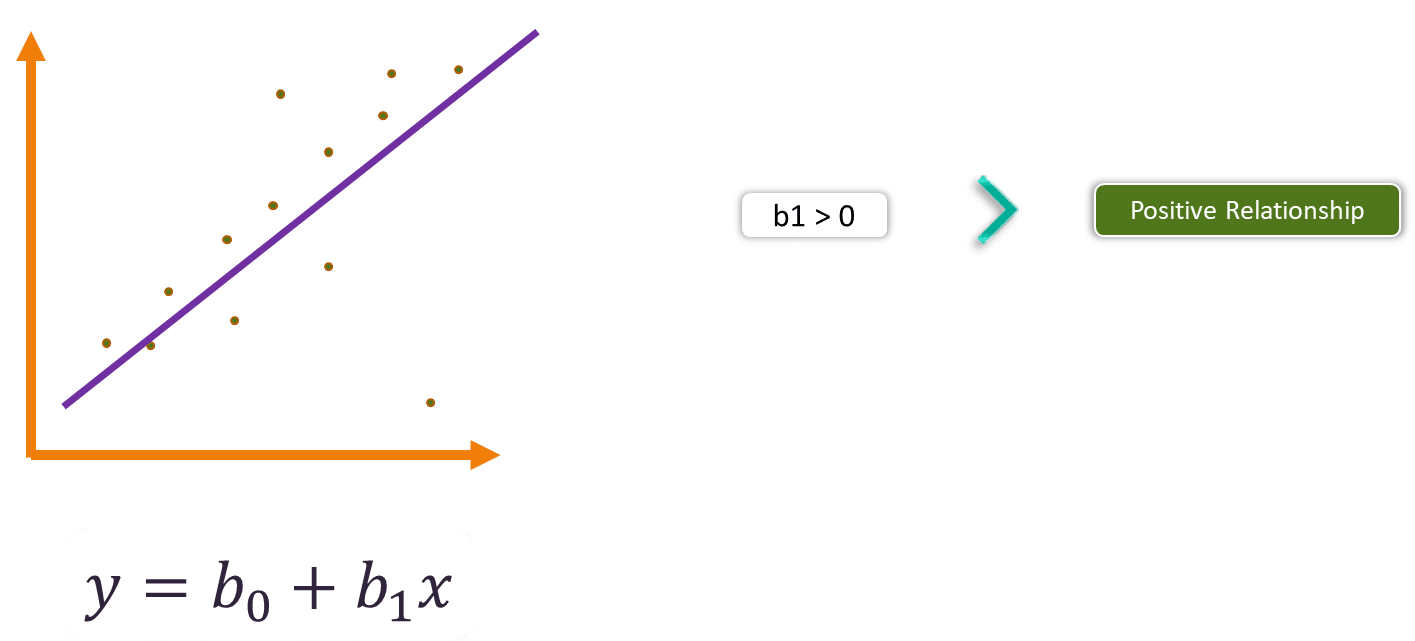
The line with the **lowest value of RSS** is the best fit line.



Now, let us see how the coefficient of x influences the relationship between the independent and the dependent variables.

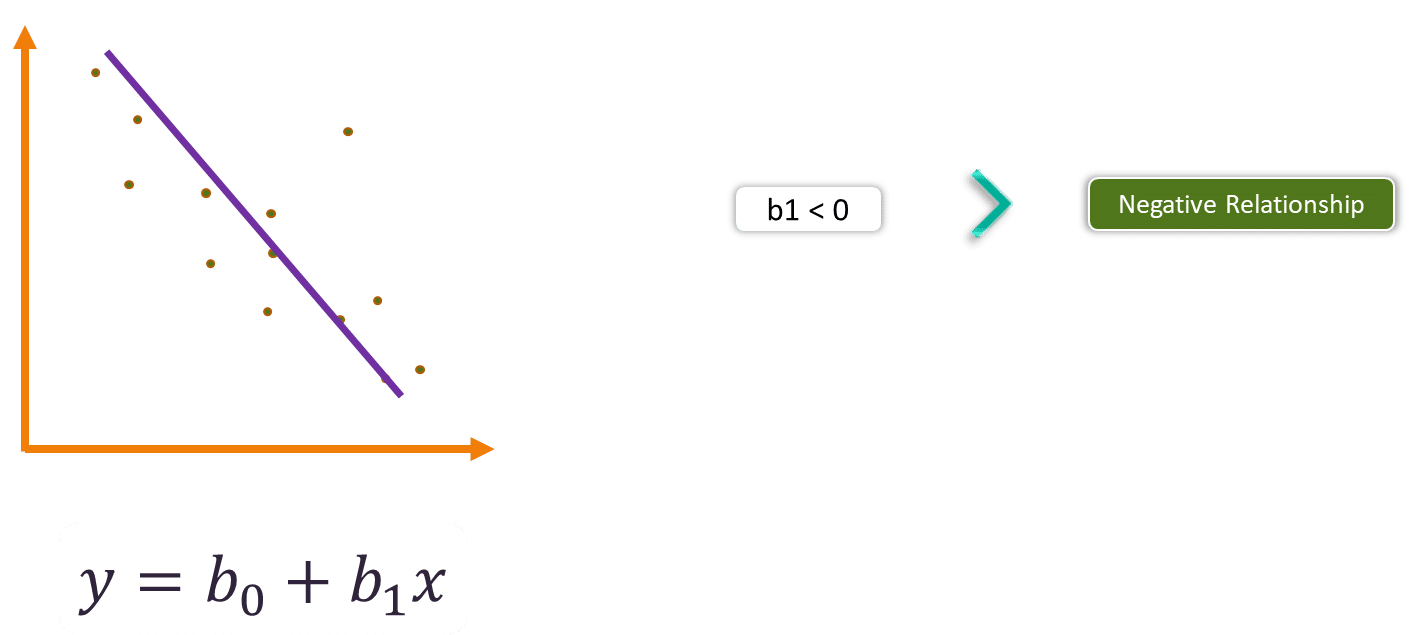
## **Regression CoefficientCoefficient**

In simple linear regression, if the coefficient of x is positive, then we can conclude that the relationship between the independent and the dependent variables is positive.



Here, if the value of x increases, the value of y also increases.

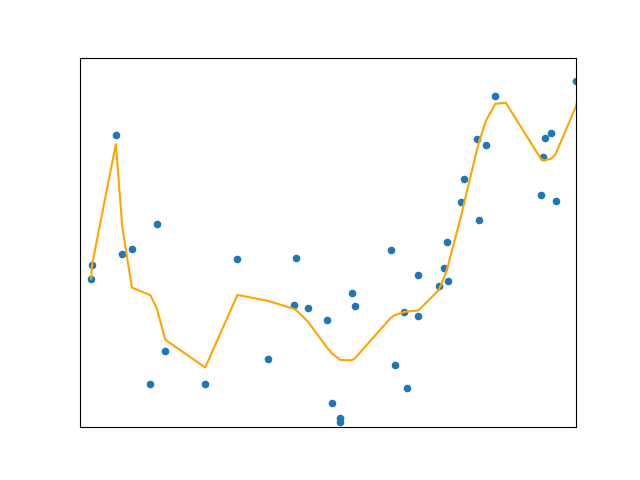
Now, if the coefficient of x is negative, then we can say that the relationship between the independent and the dependent variables is negative.



Here, if the value of x increases, the value of y decreases.

Now, let us see how we can apply these concepts to build linear regression models. In the below given Python Linear Regression Examples, we will be building two machine learning models for simple and multiple linear regression. Let’s begin.

*Linear regression requires the relation between the dependent variable and the independent variable to be linear. What if the distribution of the data was more complex as shown in the below figure? Can linear models be used to fit non-linear data? How can we generate a curve that best captures the data as shown below?*

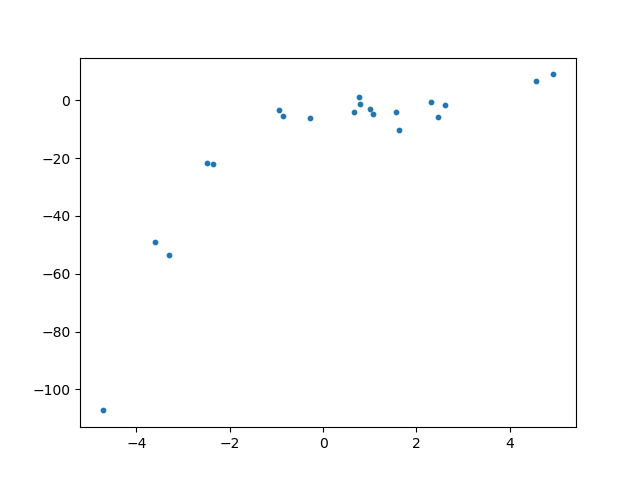


Why Polynomial Regression

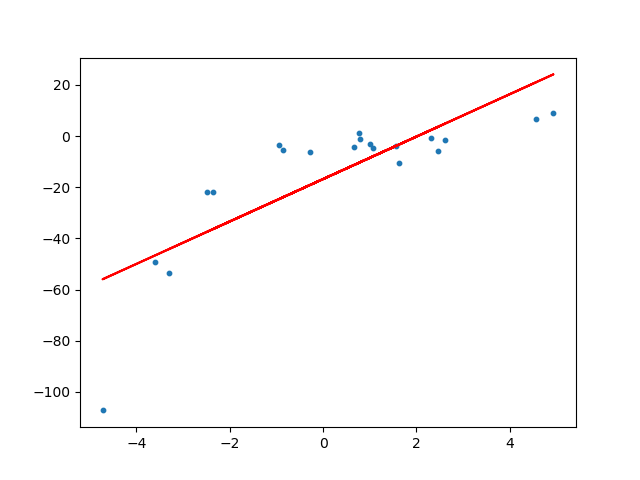
* Over-fitting vs Under-fitting

## **Why Polynomial Regression?**

To understand the need for polynomial regression, let’s generate some random dataset first.

**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**

We can see that the straight line is unable to capture the patterns in the data. This is an example of **under-fitting**. Computing the RMSE and R²-score of the linear line gives:

RMSE of linear regression is **15.908242501429998**.

R2 score of linear regression is **0.6386750054827146**

***To overcome under-fitting, we need to increase the complexity of the model.***

To generate a higher order equation we can add powers of the original features as new features. The linear model,

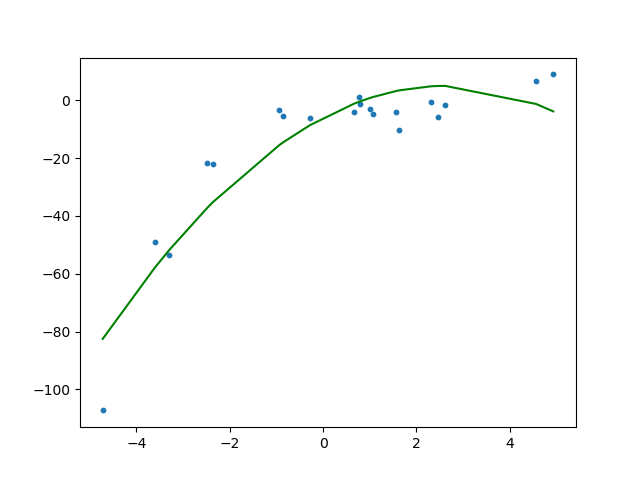
can be transformed to

 ***linear model****as the coefficients/weights associated with the features are still linear. x² is only a feature. However the curve that we are fitting is****quadratic***

To convert the original features into their higher order terms we will use the PolynomialFeatures class provided by scikit-learn. Next, we train the model using Linear Regression.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Fitting a linear regression model on the transformed features gives the below plot.



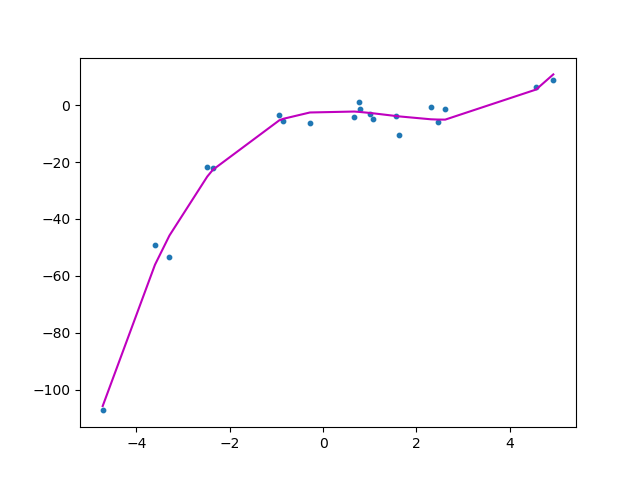
It is quite clear from the plot that the quadratic curve is able to fit the data better than the linear line. Computing the RMSE and R²-score of the quadratic plot gives:

RMSE of polynomial regression is **10.120437473614711**.

R2 of polynomial regression is **0.8537647164420812**.

***We can see that RMSE has decreased and R²-score has increased as compared to the linear line.***

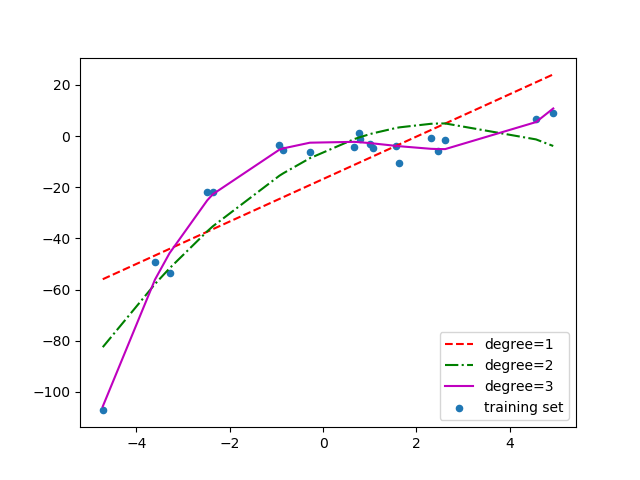
If we try to fit a cubic curve (degree=3) to the dataset, we can see that it passes through more data points than the quadratic and the linear plots.

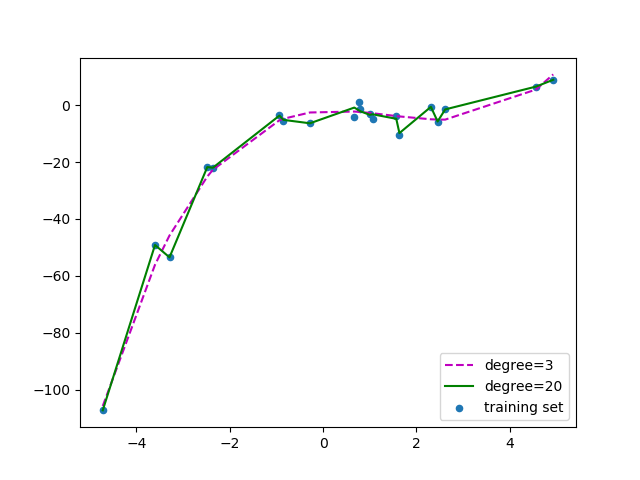
The metrics of the cubic curve is

RMSE is **3.449895507408725**

R2 score is **0.9830071790386679**

Below is a comparison of fitting linear, quadratic and cubic curves on the dataset.

If we further increase the degree to 20, we can see that the curve passes through more data points. Below is a comparison of curves for degree 3 and 20.

For degree=20, the model is also capturing the noise in the data. This is an example of **over-fitting**. Even though this model passes through most of the data, it will fail to generalize on unseen data.

***To prevent over-fitting, we can add more training samples so that the algorithm doesn’t learn the noise in the system and can become more generalized.****( Note: adding more data can be an issue if the data is itself noise).*