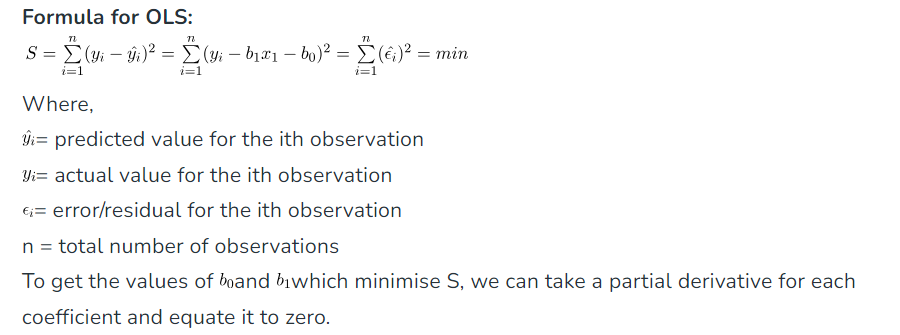
**Variations of Linear Regression:**

**Ordinary Least Squares (OLS) using statsmodels**

**Ordinary Least Squares**(**OLS**) method of linear regression.  
**Introduction :**   
A linear regression model establishes the relation between a dependent variable(**y**) and at least one independent variable(**x**) as :   
  
In *OLS* method, we have to choose the values of  b1 and b0 such that, the total sum of squares of the difference between the calculated and observed values of y, is minimized.



The **Ordinary Least Squares (OLS)** method is one of the most commonly used techniques for regression analysis.

**Ordinary least squares (OLS)** is a [linear regression](https://vitalflux.com/linear-regression-real-life-example/) technique used to find the best-fitting line for a set of data points by minimizing the residuals (the differences between the observed and predicted values). It does so by estimating the coefficients of a linear regression model by minimizing the sum of the squared differences between the observed values of the dependent variable and the predicted values from the model. It is a popular method because it is easy to use and produces decent results.

The ordinary least squares (OLS) method can be defined as a linear regression technique that is used to estimate the unknown parameters in a model. The method relies on minimizing the sum of squared residuals between the actual (observed values of the dependent variable) and predicted values from the model. The residual can be defined as the difference between the actual value and the predicted value. Another word for residual can be error. The sum of the squared differences is also known as the residual sum of squares (RSS). The OLS method minimizes the RSS by finding the values of the coefficients that result in the smallest possible RSS. The resulting line is called the regression line, which represents the best fit for the data.

In mathematical terms, this can be written as:

**Minimize**∑(**yi** – **ŷ**i)**^2**

where **y**i is the actual value, **ŷ**i is the predicted value. A linear regression model used for determining the value of the response variable, **ŷ,** can be represented as the following equation.

**y = b0 + b1x1 + b2x2 + … + bnxn + e**

where:

* y is the dependent variable
* b0 is the intercept
* b1, b2, …, bn are the coefficients of the independent variables x1, x2, …, xn
* e is the error term

The coefficients b1, b2, …, bn can also be called the **coefficients of determination**. The goal of the OLS method can be used to estimate the unknown parameters (**b1, b2, …, bn**) by minimizing the sum of squared residuals (RSS). The sum of squared residuals is also termed the sum of squared error (SSE).

This method is also known as the **least-squares method** for regression or linear regression.

### **Assumptions of OLS**

The OLS method relies on several assumptions to be valid. The following is the list of key assumptions:

1. **Linearity**: There must be linear relationship between the dependent variable and the independent variables.
2. **Independence**: The observations must be independent of each other.
3. **Homoscedasticity**: The variance of the residuals should be constant across all levels of the independent variables.
4. **Normality**: The residuals / errors should be normally distributed.
5. **No multicollinearity**: The independent variables should not be highly correlated with each other.

## **Minimizing the sum of squares residuals using the calculus method**

The following represents the calculus method for minimizing the sum of squares residuals to find the unknown parameters for the model y = mx + b

Take the partial derivative of the cost function, sum of squared residuals, **∑(yi – ŷi)^2**with respect to m:

**∂/∂m (SSE) = ∑-2Xi(yi – ŷi)**

Take the partial derivative of the cost function,**∑ (yi – ŷi)^2** with respect to b:

**∂/∂b (SSE) = ∑-2(yi – ŷi)**

Set the partial derivatives equal to zero and solve for m and b:

**∑-2Xi(yi – ŷi) = 0**

**∑-(yi – ŷi) = 0**

This results in the following two equations:

**∑yi\*xi = m∑xi\*xi + b\*∑xi**

**∑yi = m∑xi + b\*n**

where n is the number of data points. These two equations can be solved simultaneously to find the values for m and b. Let’s say that the following three points are available such as (3, 7), (4, 9), (5, 12). And, the ask is to find the best fit line.

We will apply the calculus technique and use the above formulas. We will use the following formula:

**∑-2Xi(yi – ŷi) = 0**

The following calculation will happen:

-2[3(7 – (3m + b)) + 4(9 – (4m + b)) + 5(12 – (5m + b))] = 0

=> 3\*7 + 4\*9 + 5\*12 – (9m + 3b + 16m + 4b + 25m + 5b) = 0

=> 21 + 36 + 60 – (50m + 12b) = 0

=> **116 = 50m + 12b …. eq (1)**

Let’s use another formula to find another equation:

**∑-(yi – ŷi) = 0**

The following calculation will happen:

7 – (3m + b) + 9 – (4m + b) + 12 – (5m + b) = 0

=> **28 = 12m + 3b … eq(2)**

**The above two equations can be solved and the values of m and b can be found.**

**LOWESS Regression.**

Since LOWESS is a **non-parametric** fitting technique, you do not need to assume that data follows any specific distribution. However, at the same time, non-parametric fitting means that at the end, you will not have a global equation for you to use to predict the values of new data points. However there is another way of doing predictions using statsmodels in python.

Given the above advantages and disadvantages, LOWESS is often used to perform the following analysis:

* A regression analysis where simple linear regression cannot produce a line of good fit due to data attributes not following a linear relationship. Note, a special case of multiple linear regression known as polynomial regression could also be used in this case.
* Fitting a line to a scatter plot or time plot where noisy data values, sparse data points, or weak interrelationships interfere with your ability to see a line of best fit.

It is a good way to model a relationship between two variables that do not fit a predefined distribution and have a non-linear relationship.

Said that one important addition to LOWESS over OLS is that it applies weightings as you might have guessed from the algorithm’s name — Locally **Weighted** Scatterplot Smoothing.

Typically, the algorithm uses a tri-cube weight function (see below), although other functions can also be used.

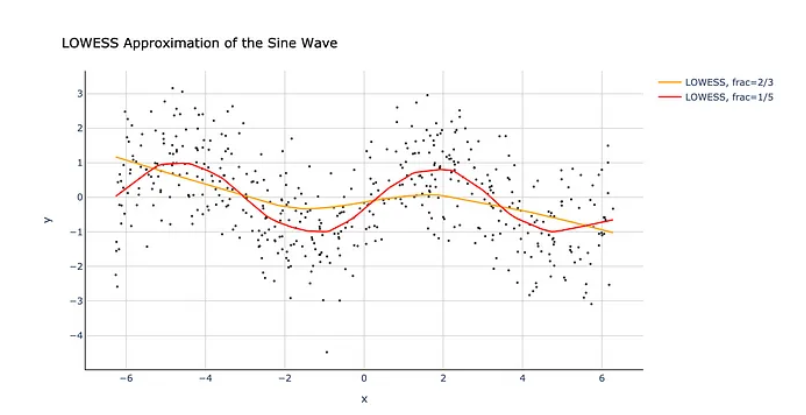
**Weight functionw(x) = (1 - |d|³)³**, where d is the distance of a given data point from the point on the curve being fitted, scaled to lie in the range between 0 and 1

The weight function gives the most weight to the data points nearest to the point of estimation and the least weight to the data points that are furthest away. It is based on the idea that points near each other in the explanatory variable space are more likely to be related to each other in a simple way than points that are further apart.

## Intuitive explanation

We will use an example to illustrate how LOWESS works. Let us start by creating a scatterplot where the data points follow a sine wave pattern, but they have some random noise added, making the pattern less obvious.

After that, we perform a LOWESS regression analysis a couple of times using different hyperparameters and add LOWESS curves to the plot:

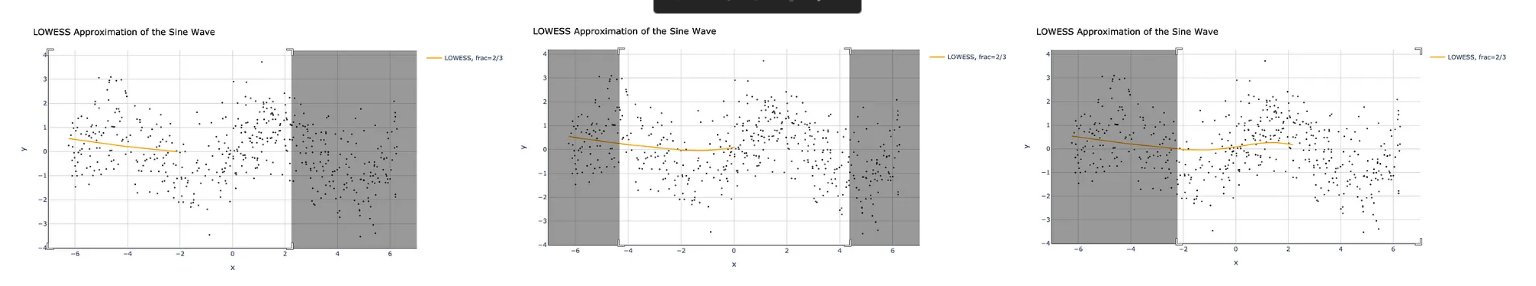


As you can see, there is a lot of noise in the data, with the relationship between x and y being non-linear (periodic, in fact). Clearly, simple linear regression would not give us a meaningful result here since it can only produce a straight line (**unlike LOWESS or polynomial regression).**

The first thing that LOWESS does is selects a subset of the data. We can control the size of the subsets by specifying the ‘fraction’ hyperparameter. The two lines that we plotted above have hyperparameter values of 2/3 and 1/5.

Let’s take the orange line as an example, with a default hyperparameter value of 2/3. This means that the LOWESS algorithm takes a subset of two-thirds of the entire data and performs a **weighted linear least squares regression** over the span of that data while moving by one point at a time and repeating the same process until it has gone through every single point.

Starting from the very left and going right one point at a time, we can see how the orange line remains straight until x=-2.1. This is because the entire data ranges from -2π to +2π (-6.28 to +6.28) with an initial two-thirds window ranging from -6.28 to +2.1. Hence, only when the algorithm goes beyond -2.1 (midpoint between -6.28 and +2.1), it starts encountering new data points within its window, changing the curve's slope.



f we now look at the red line (frac=1/5), we can see how narrowing the window to one-fifth of the data makes the algorithm **more sensitive** towards local trends. The featured gif image at the beginning of this story illustrates how this red line (frac=1/5) moves across the data performing weighted linear regression. This is the same process as described for the orange line (frac=2/3) except, the window the algorithm sees is narrower.

**While in this example, making the window smaller helped us get a better approximation of the sine wave, it is not always desirable to make a ‘fraction’ hyperparameter smaller. This largely depends on the data you are analyzing, as making a window smaller runs a risk of overfitting:**

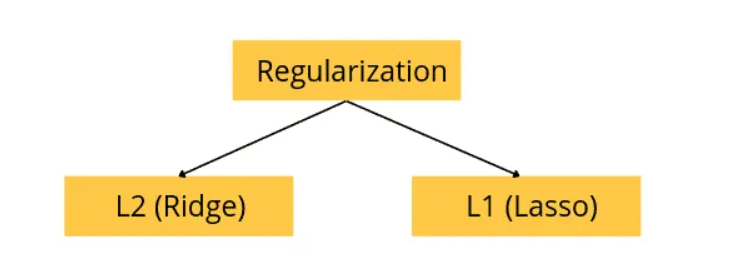
Simple linear regression has only one slope parameter meaning that it has the same steepness of the curve throughout. Meanwhile, LOWESS can adjust the curve's steepness at various points, producing a better fit than that of simple linear regression.

# Conclusion

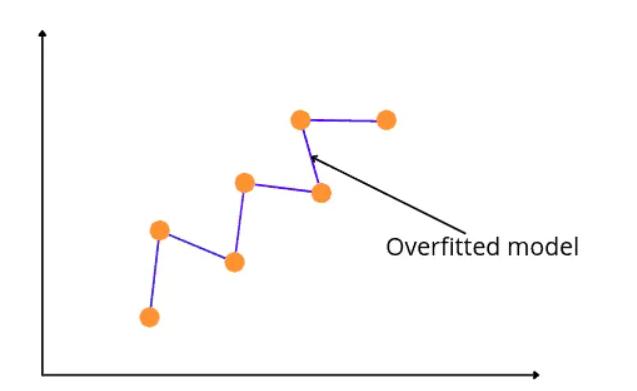
LOWESS is not something that you may want to use in all of your regression models as it follows a non-parametric approach and is quite computationally intensive. However, **it is a good way to model a relationship between two variables that do not fit a predefined distribution and have a non-linear relationship.**

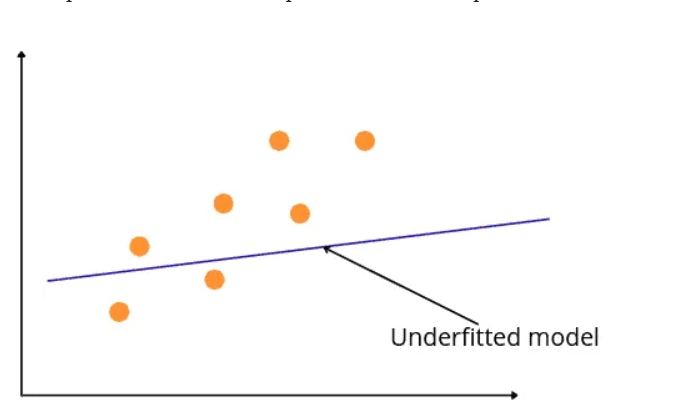
## **Ridge Regression and Lasso Regression**

Ridge and Lasso Regressions are the two most common types of regularization techniques in regression methods. They are also known as L1 (the Lasso regression) and L2 (Ridge regression) regularizations.



if we allow the model to train on the dataset many times, the model will find a lot of patterns, including unnecessary and noisy ones. The model will then learn well on the training dataset and fit very well, but then it will not be able to predict other datasets. So, in simple words, a scenario where the model learns from the data’s details and noise and tries to fit each data point on the curve is called **overfitting**.





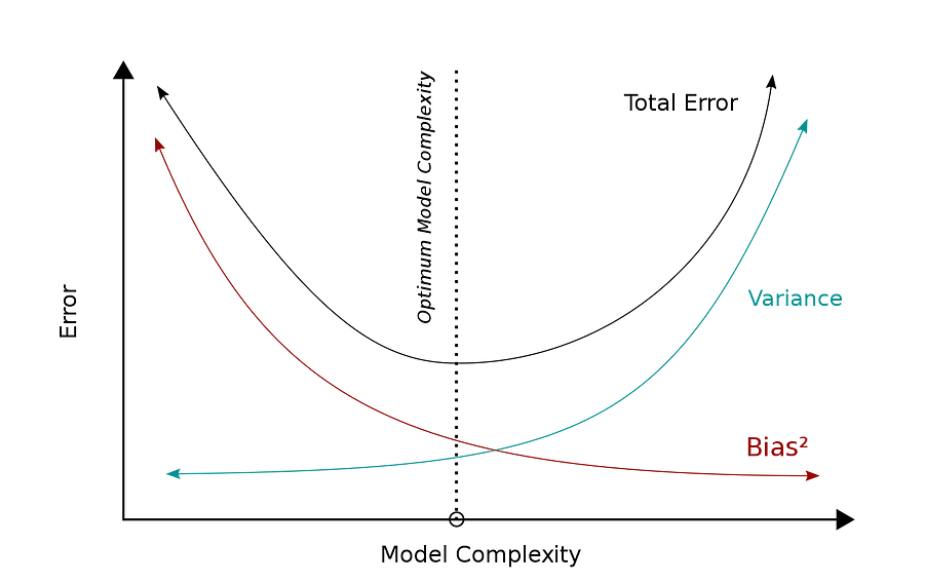
### What Is the Difference Between Bias And Variance?

A Bias occurs in a model when the model has limited flexibility to learn from the dataset. Such models pay very little attention to the training dataset and can’t find the trends. This type of modeling always leads to a high error in training and testing datasets. In other words, we can say that high bias causes underfitting in the model.

While variance defines the model’s sensitivity to a specific dataset, a model with a high variance pays a lot of attention to the training dataset and does not generalize; therefore, the prediction error is high. Such models usually perform well on training datasets but give high error rates on the testing datasets. In other words, high variance leads to the overfitting of the model.

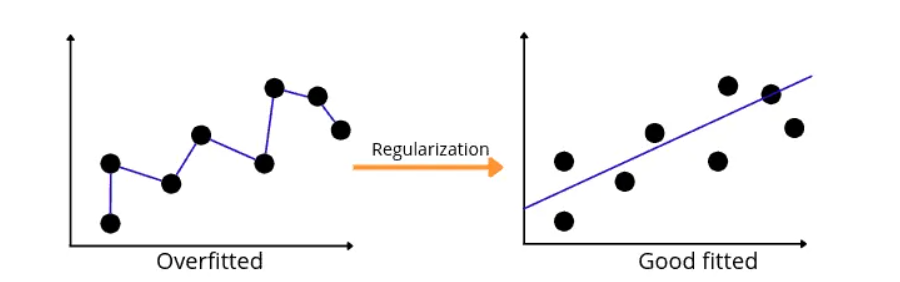
An optimal model is one in which the model is sensitive to the pattern in our model but at the same time can generalize to new data. This happens when Bias and Variance are both optimal. **We can achieve such an optimal model in over or under-fitted models by using Regularization (Lasso and Ridge regression).**

**We need to find a balance between bias and variance as a tradeoff, since variance tends to overfit while bias tends to underfit a model:**

****

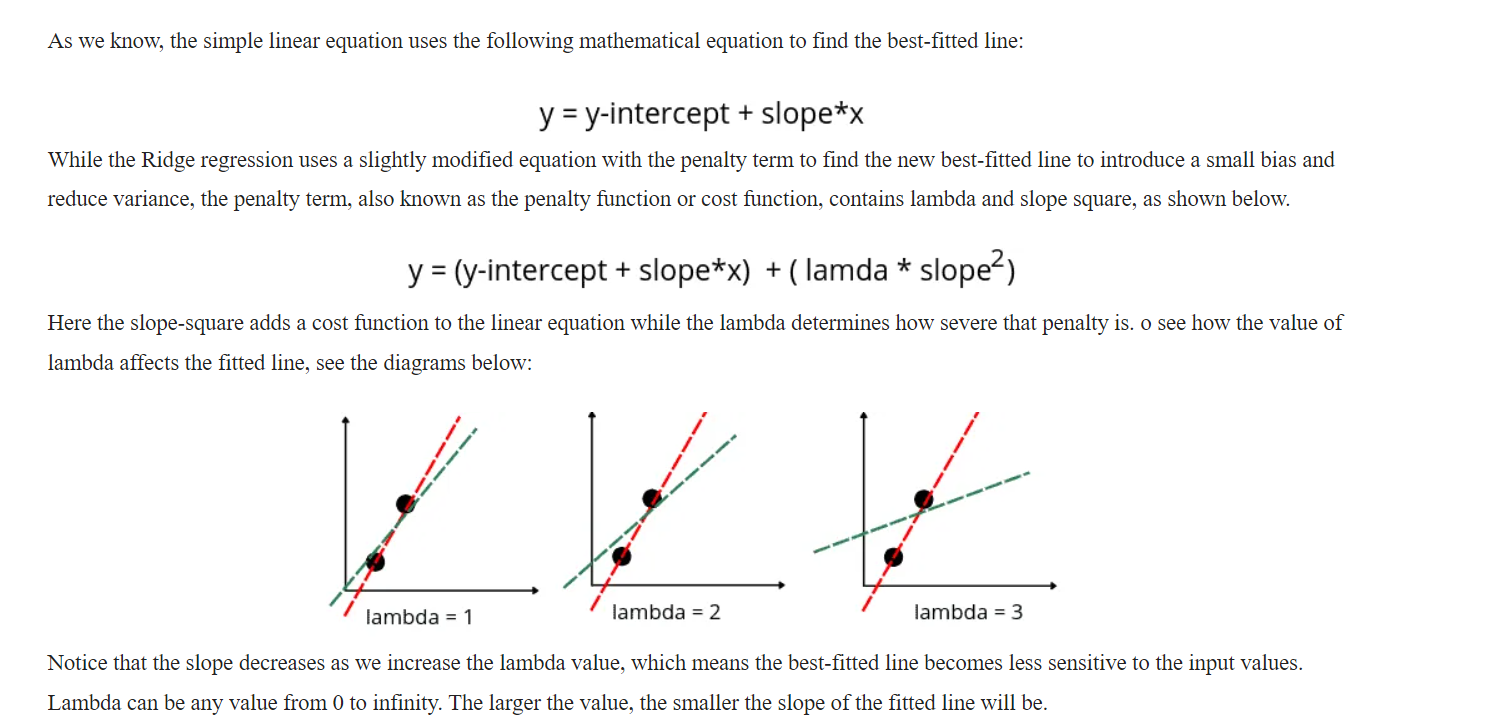
### What Is Regularization In Machine Learning?

Regularization refers to techniques used to calibrate machine learning models to minimize the adjusted loss function and prevent overfitting or underfitting. Using regularization, we can fit our machine learning model appropriately on a given test set and reduce its errors.



****

The main goal of Ridge regression is to find a new line that doesn’t fit well on the training data but reduces the error on the testing data. In other words, we will introduce a small amount of bias for the new fitted line on the training dataset to get a low variance for the testing data. For example, see the new fitted line (ridge regression) below, which reduces the variance compared to the previous overfitted line.

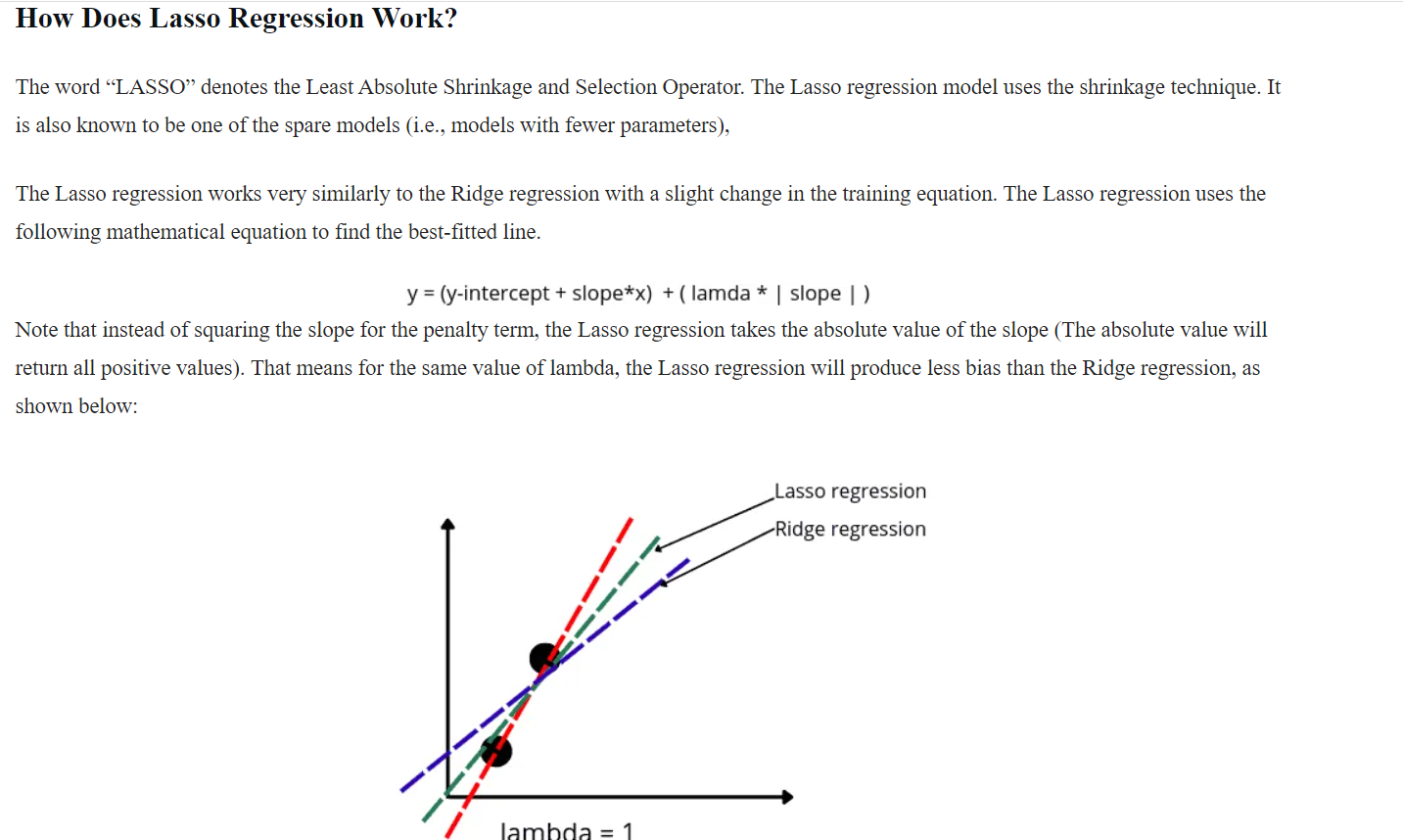
****

Ridge regression shrinks the regression coefficients, so that variables (predictors), with minor contribution to the outcome, have their coefficients close to zero. The shrinkage of the coefficients is achieved by penalizing the regression model with a penalty term called **L2-norm**, which is the sum of the squared coefficients. The amount of the penalty can be fine-tuned using a constant called lambda (λ). Selecting a good value for λ is critical.

When λ=0, the penalty term has no effect, and ridge regression will produce the classical least square coefficients. However, as λ increases to infinite, the impact of the shrinkage penalty grows, and the ridge regression coefficients will get close zero. The larger the lambda, the smaller the slope of the fitted line will be.

**One important advantage of the ridge regression, is that it still performs well, compared to the ordinary least square method, in a situation where you have a large multivariate data with the number of predictors (p) larger than the number of observations (n).**

Ridge regression shrinks the coefficients towards zero, **but it will not set any of them exactly to zero**. The lasso regression is an alternative that overcomes this drawback.

****

Since Lasso does not square the slope and use the absolute value, it has less bias than Ridge and hence more variance.

One key difference between L1 and L2 regularization is their impact on the output of the model. **L1 regularization tends to set some parameter coefficients to zero,** leading to sparser models. On the other hand, L2 regularization encourages small, non-zero parameter coefficients, resulting in models with less sparsity.

For the ease of understanding, I'll describe this using an example. Let's say that you are collecting data from a device which has 12 sensors. And you have collected data for 10 days. **This is called sparse data because most of the sensor outputs are zero.**

L1 regularization penalizes the sum of absolute values of the weights, whereas L2 regularization penalizes the sum of squares of the weights. The L1 regularization solution is sparse. The L2 regularization solution is non-sparse.

**The standard linear model (or the ordinary least squares method) performs poorly in a situation, where you have a large multivariate data set containing a number of variables superior to the number of samples.** More columns than rows.

**From a practical standpoint, L1 tends to shrink coefficients to zero whereas L2 tends to shrink coefficients evenly. L1 is therefore useful for feature selection, as we can drop any variables associated with coefficients that go to zero. L2, on the other hand, is useful when you have collinear/codependent features.**

### **Lasso regression**

It shrinks the regression coefficients toward zero by penalizing the regression model with a penalty term called **L1-norm**, which is the sum of the absolute coefficients. In the case of lasso regression, the penalty has the effect of forcing some of the coefficient estimates, with a minor contribution to the model, to be exactly equal to zero. This means that, lasso can be also seen as an alternative to the subset selection methods for performing variable selection in order to reduce the complexity of the model.

As in ridge regression, selecting a good value of λ for the lasso is critical.

One obvious advantage of lasso regression over ridge regression, is that it produces simpler and more interpretable models that incorporate only a reduced set of the predictors. However, neither ridge regression nor the lasso will universally dominate the other.

**Generally, lasso might perform better in a situation where some of the predictors have large coefficients, and the remaining predictors have very small coefficients. It eliminates the ones with small coefficients, simplifying the models.**

**Ridge regression will perform better when the outcome is a function of many predictors, all with coefficients of roughly equal size. Ridge regression diminishes the coefficient of features with less impact but never reduces to zero compared to lasso regression.**

### **Elastic Net regularization**

Elastic Net produces a regression model that is penalized with both the **L1-norm** and **L2-norm**. The consequence of this is to effectively shrink coefficients (like in ridge regression) and to set some coefficients to zero (as in LASSO). Therefore, Elastic Net is a combination of both Lasso and Ridge regression.

**Confusingly,** the alpha hyperparameter can be set via the “l1\_ratio” argument that controls the contribution of the L1 and L2 penalties and the lambda hyperparameter can be set via the “alpha” argument that controls the contribution of the sum of both penalties to the loss function. Example:

# define model

model = ElasticNet(alpha=1.0, l1\_ratio=0.5)

# All regression models assume that independent variables are not correlated that means that a variation in one independent variable does not affect the other independent variables values

# However, it is good for our models that indendent variables and the target, dependent variable, that we want to predict have high correlation.