What Are Overfitting and Underfitting?

To train our machine learning model, we give it some data to learn from. The process of plotting a series of data points and drawing the best fit line to understand the relationship between the variables is called Data Fitting. Our model is the best fit when it can find all necessary patterns in our data and avoid the random data points and unnecessary patterns called Noise.

f we allow our machine learning model to look at the data too many times, it will find a lot of patterns in our data, including the ones which are unnecessary. It will learn really well on the test dataset and fit very well to it. It will learn important patterns, but it will also learn from the noise in our data and will not be able to predict on other datasets.

A scenario where the machine learning model tries to learn from the details along with the noise in the data and tries to fit each data point on the curve is called Overfitting.

In the figure depicted below, we can see that the model is fit for every point in our data. If given new data, the model curves may not correspond to the patterns in the new data, and the model cannot predict very well in it.

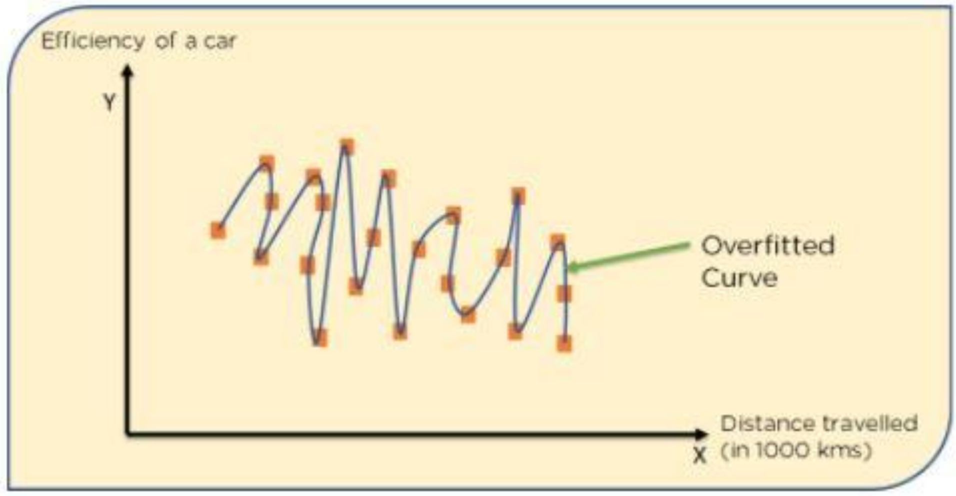


Figure 1: Overfitted Model

Conversely, in a scenario where the model has not been allowed to look at our data a sufficient number of times, the model won't be able to find patterns in our test dataset. It will not fit properly to our test dataset and fail to perform on new data too.

A scenario where a machine learning model can neither learn the relationship between variables in the testing data nor predict or classify a new data point is called Underfitting.

The below diagram shows an under-fitted model. We can see that it has not fit properly to the data given to it. It has not found patterns in the data and has ignored a large part of the dataset. It cannot perform on both known and unknown data.

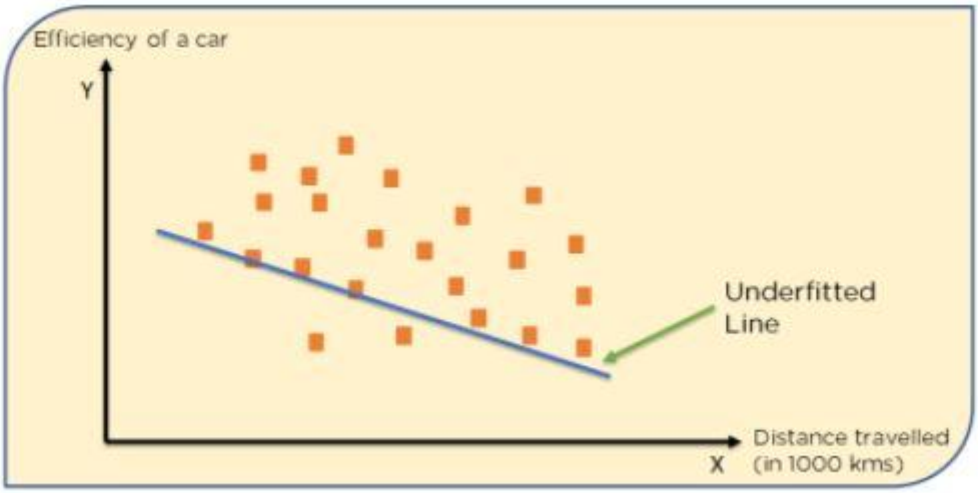


Figure 2: Underfitted Model

What are Bias and Variance?

A Bias occurs when an algorithm has limited flexibility to learn from data. Such models pay very little attention to the training data and oversimplify the model therefore the validation error or prediction error and training error follow similar trends. Such models always lead to a high error on training and test data. High Bias causes underfitting in our model.

Variance defines the algorithm's sensitivity to specific sets of data. A model with a high variance pays a lot of attention to training data and does not generalize therefore the validation error or prediction error are far apart from each other. Such models usually perform very well on training data but have high error rates on test data. High Variance causes overfitting in our 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 call this Bias-Variance Tradeoff and we can achieve it in over or under fitted models by using Regression.

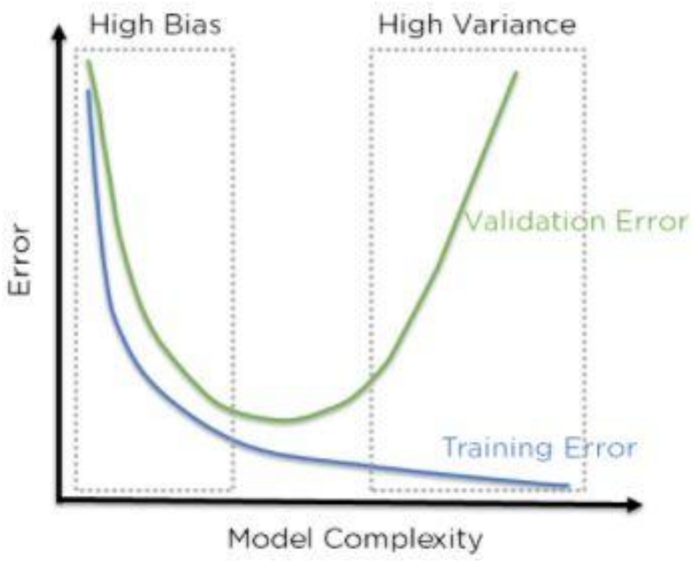


Figure 3: Error in testing and training datasets with high bias and variance

In the above figure, we can see that when bias is high, the error in both testing and training set is also high. When Variance is high, the model performs well on our training set and gives a low error, but the error in our testing set is very high. In the middle of this exists a region where the bias and variance are in perfect balance to each other, and here, but the training and testing errors are low.

\What is Regularization in Machine Learning?

Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

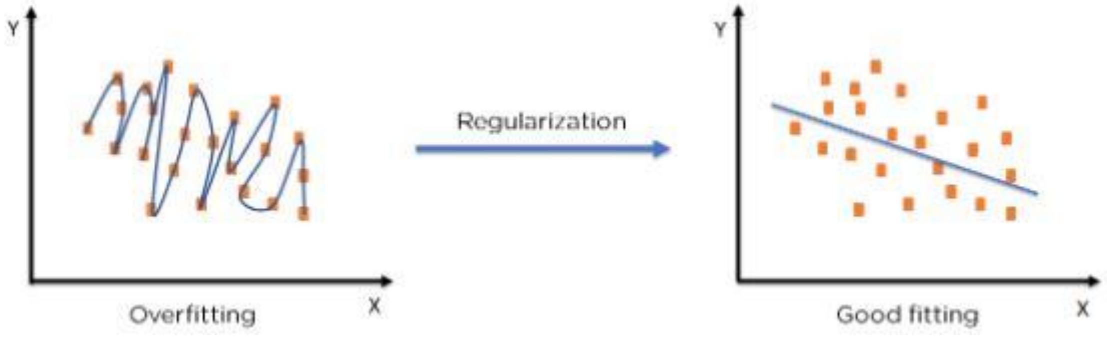


Figure 5: Regularization on an over-fitted model

Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.

Regularization Techniques

There are two main types of regularization techniques: Ridge Regularization and Lasso Regularization.

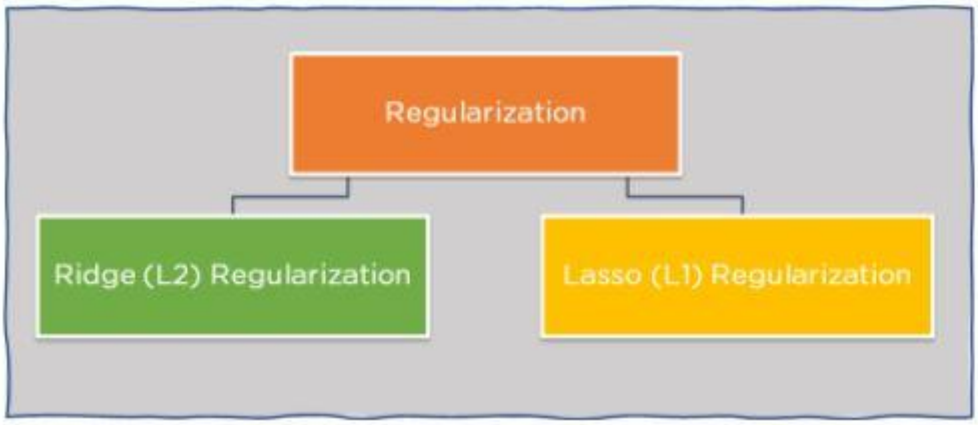
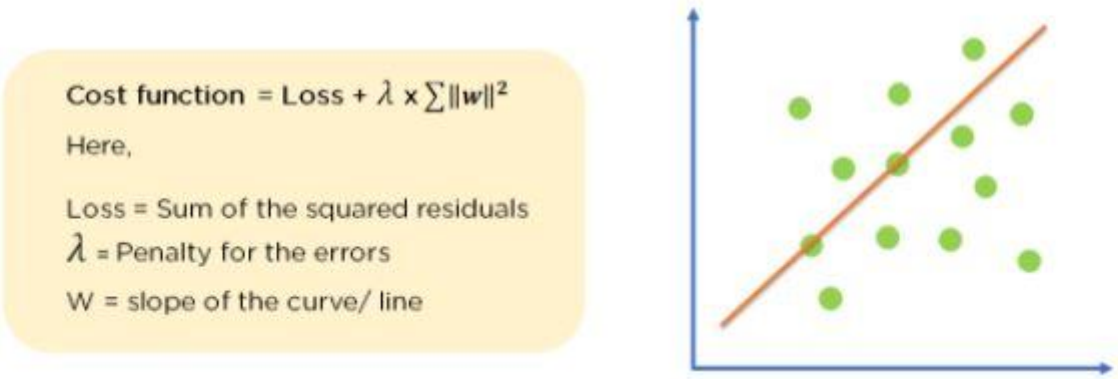


Figure 6: Regularization techniques

Ridge Regularization :

Also known as Ridge Regression, it modifies the over-fitted or under fitted models by adding the penalty equivalent to the sum of the squares of the magnitude of coefficients.

This means that the mathematical function representing our machine learning model is minimized and coefficients are calculated. The magnitude of coefficients is squared and added. Ridge Regression performs regularization by shrinking the coefficients present. The function depicted below shows the cost function of ridge regression :



Cost function = Loss + λ x $\sum \|w\|^2$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

W = slope of the curve/ line

Figure 7: Cost Function of Ridge Regression

In the cost function, the penalty term is represented by Lambda λ. By changing the values of the penalty function, we are controlling the penalty term. The higher the penalty, it reduces the magnitude of coefficients. It shrinks the parameters. Therefore, it is used to prevent multicollinearity, and it reduces the model complexity by coefficient shrinkage.

Consider the graph illustrated below which represents Linear regression :

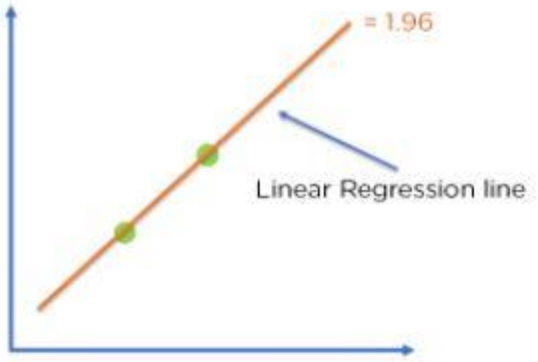


Figure 8: Linear regression model

Cost function = Loss + λ x∑‖w‖^2

For Linear Regression line, let’s consider two points that are on the line,

Loss = 0 (considering the two points on the line)

λ= 1

w = 1.4

Then, Cost function = 0 + 1 x 1.42

= 1.96

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.22 = 0.13

λ = 1

w = 0.7

Then, Cost function = 0.13 + 1 x 0.72

= 0.62

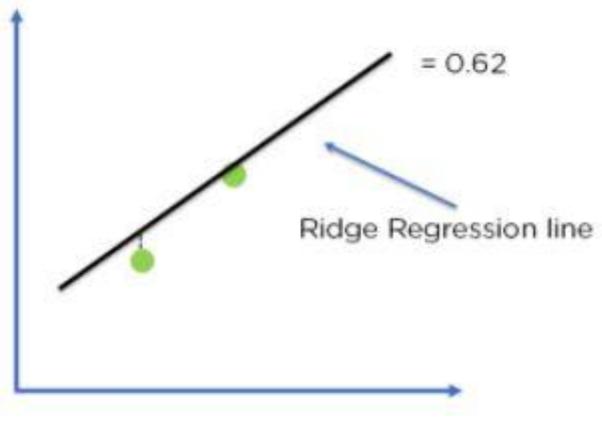


Figure 9: Ridge regression model

Comparing the two models, with all data points, we can see that the Ridge regression line fits the model more accurately than the linear regression line.

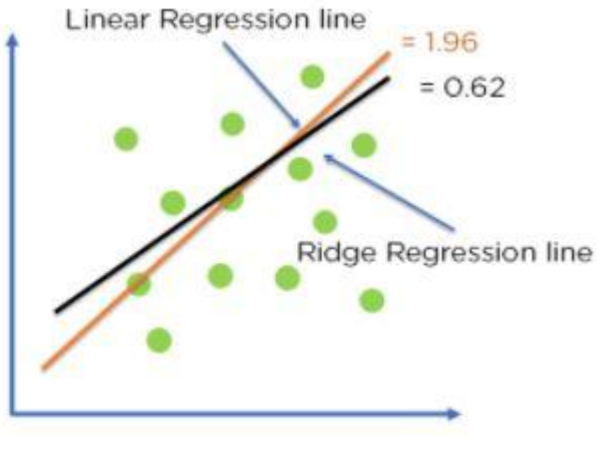
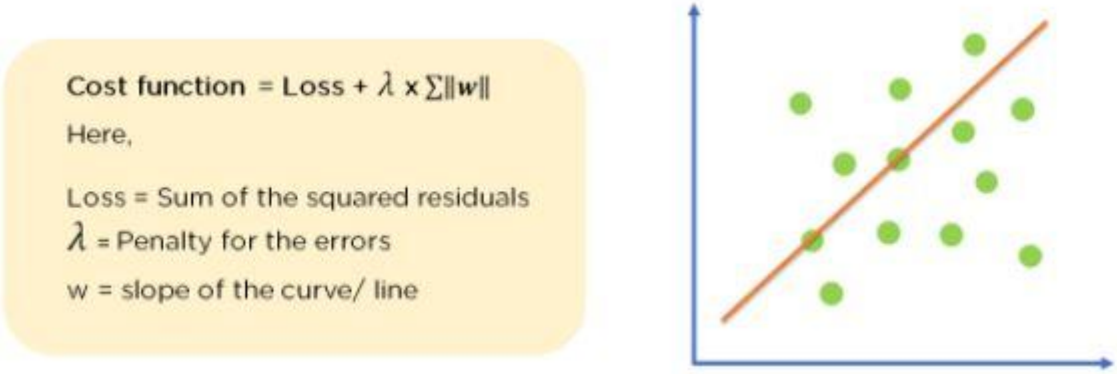


Figure 10: Optimization of model fit using Ridge Regression

Lasso Regression

It modifies the over-fitted or under-fitted models by adding the penalty equivalent to the sum of the absolute values of coefficients.

Lasso regression also performs coefficient minimization, but instead of squaring the magnitudes of the coefficients, it takes the true values of coefficients. This means that the coefficient sum can also be 0, because of the presence of negative coefficients. Consider the cost function for Lasso regression :



Cost function = Loss + λ x $\sum\|w\|$

Here,

Loss = Sum of the squared residuals

λ = Penalty for the errors

w = slope of the curve/ line

Figure 11: Cost function for Lasso Regression

We can control the coefficient values by controlling the penalty terms, just like we did in Ridge Regression. Again consider a Linear Regression model :

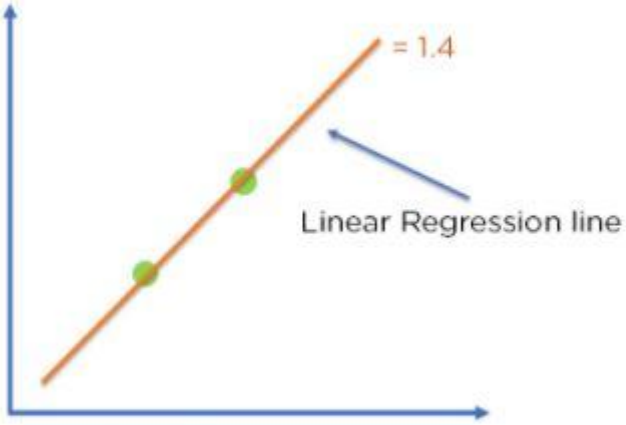


Figure 12: Linear Regression Model

Figure 12: Linear Regression Model

Cost function = Loss + λ x ∑‖w‖

For Linear Regression line, let’s assume,

Loss = 0 (considering the two points on the line)

λ = 1

w = 1.4

Then, Cost function = 0 + 1 x 1.4

= 1.4

For Ridge Regression, let’s assume,

Loss = 0.32 + 0.12 = 0.1

λ = 1

w = 0.7

Then, Cost function = 0.1 + 1 x 0.7

= 0.8

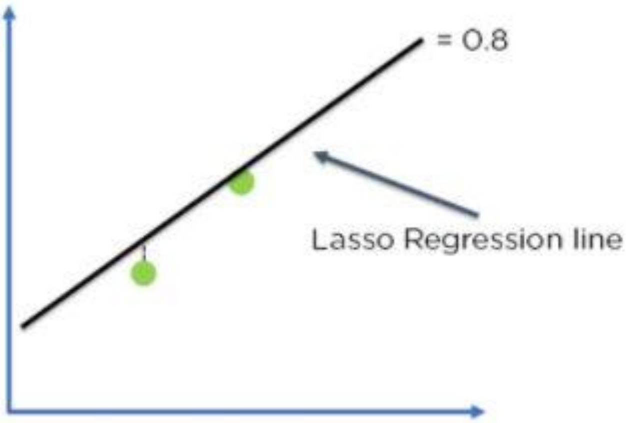


Figure 13: Lasso Regression

Comparing the two models, with all data points, we can see that the Lasso regression line fits the model more accurately than the linear regression line.