What is bias?
Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.
In general, a machine learning model analyses the data, find patterns in it and make predictions. While training, the model learns these patterns in the dataset and applies them to test data for prediction. While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias. It can be defined as an inability of machine learning algorithms such as Linear Regression to capture the true relationship between the data points. Each algorithm begins with some amount of bias because bias occurs from assumptions in the model, which makes the target function simple to learn. A model has either:

- **Low Bias:** A low bias model will make fewer assumptions about the form of the target function.
- **High Bias:** A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**

Generally, a linear algorithm has a high bias, as it makes them learn fast. The simpler the algorithm, the higher the bias it has likely to be introduced. Whereas a nonlinear algorithm often has low bias.

What is variance?
Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

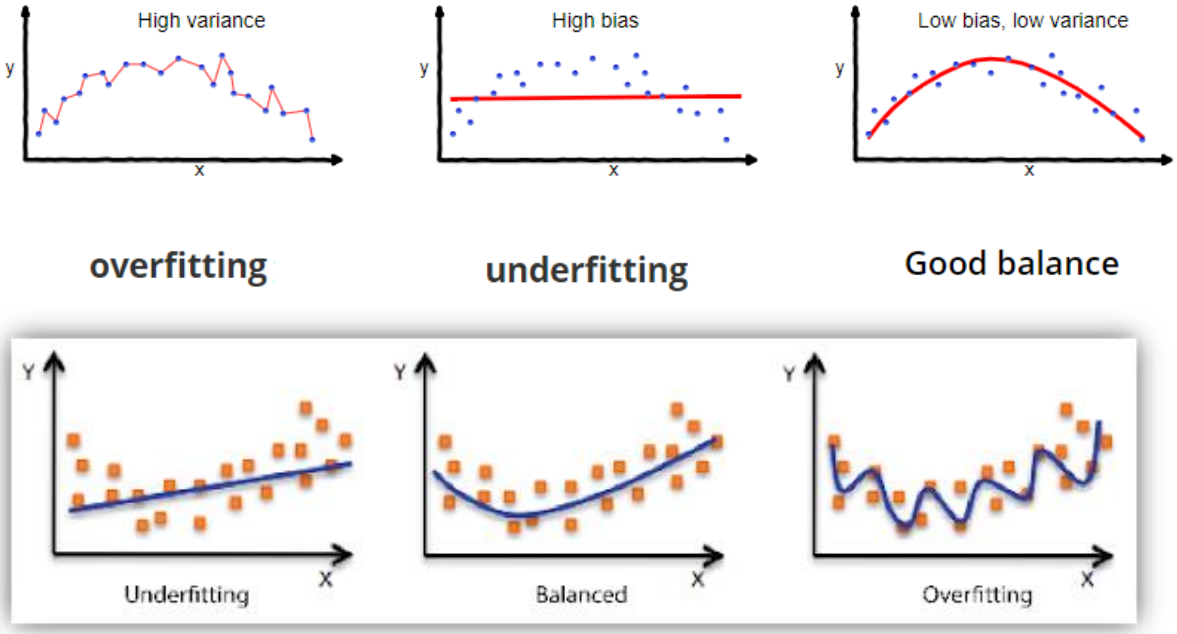
The variance would specify the amount of variation in the prediction if the different training data was used. In simple words, ***variance tells that how much a random variable is different from its expected value.*** Ideally, a model should not vary too much from one training dataset to another, which means the algorithm should be good in understanding the hidden mapping between inputs and output variables. Variance errors are either of **low variance or high variance.**

Low variance means there is a small variation in the prediction of the target function with changes in the training data set. At the same time, **High variance** shows a large variation in the prediction of the target function with changes in the training dataset.

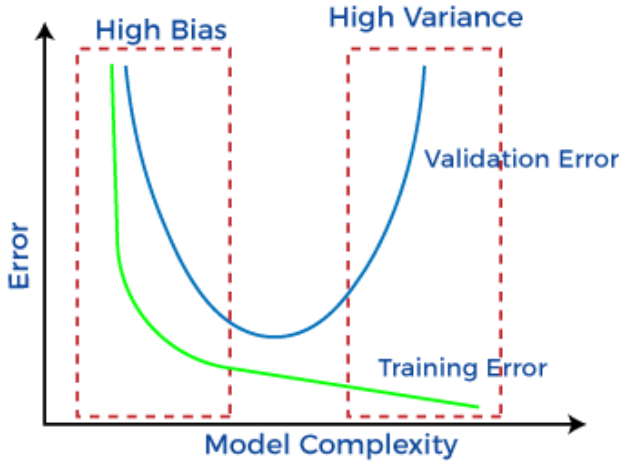
A model that shows high variance learns a lot and perform well with the training dataset, and does not generalize well with the unseen dataset. As a result, such a model gives good results with the training dataset but shows high error rates on the test dataset.
Since, with high variance, the model learns too much from the dataset, it leads to overfitting of the model. A model with high variance has the below problems:

- A high variance model leads to overfitting.
- Increase model complexities.

Usually, nonlinear algorithms have a lot of flexibility to fit the model, have high variance.



Machine learning is a branch of Artificial Intelligence, which allows machines to perform data analysis and make predictions. However, if the machine learning model is not accurate, it can make predictions errors, and these prediction errors are usually known as Bias and Variance. In machine learning, these errors will always be present as there is always a slight difference between the model predictions and actual predictions. The main aim of ML/data science analysts is to reduce these errors in order to get more accurate results. In this topic, we are going to discuss bias and variance, Bias-variance trade-off, Underfitting and Overfitting. But before starting, let's first understand what errors in Machine learning are?



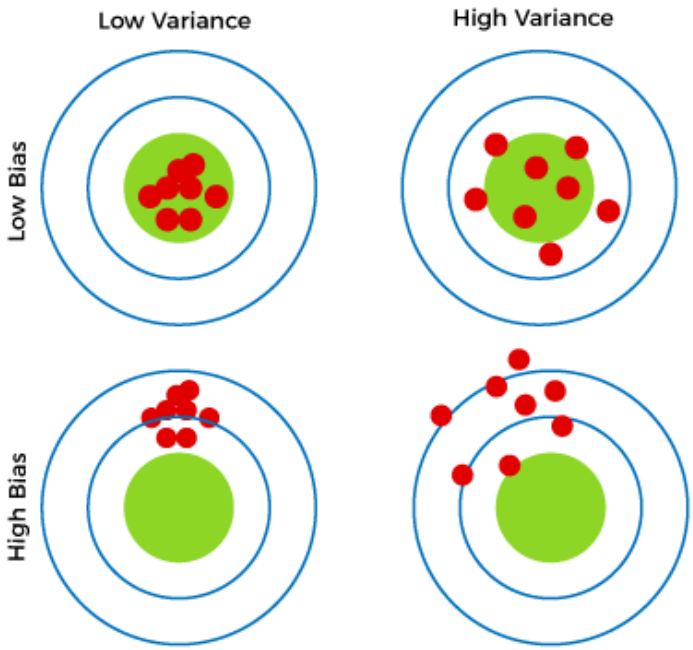
Different Combinations of Bias-Variance

There are four possible combinations of bias and variances, which are represented by the below diagram:

1. **Low-Bias,Low-Variance:**
   The combination of low bias and low variance shows an ideal machine learning model. However, it is not possible practically.
2. **Low-Bias, High-Variance:** With low bias and high variance, model predictions are inconsistent and accurate on average. This case occurs when the model learns with a large number of parameters and hence leads to an **overfitting**
3. **High-Bias, Low-Variance:** With High bias and low variance, predictions are consistent but inaccurate on average. This case occurs when a model does not learn well with the training

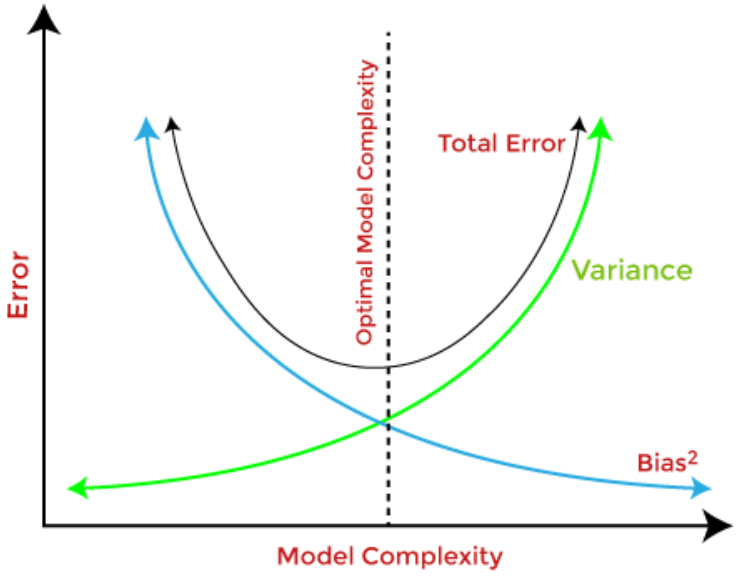
dataset or uses few numbers of the parameter. It leads to **underfitting** problems in the model.

4. **High-Bias, High-Variance:**
   With high bias and high variance, predictions are inconsistent and also inaccurate on average.



Bias-Variance Trade-Off

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off.**



For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

- If we decrease the variance, it will increase the bias.
- If we decrease the bias, it will increase the variance.

Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a high variance algorithm may perform well with training data, but it may lead to overfitting to noisy data. Whereas, high bias algorithm generates a much simple model that may not even capture important regularities in the data. So, we need to find a sweet spot between bias and variance to make an optimal model.

Hence, the ***Bias-Variance trade-off is about finding the sweet spot to make a balance between bias and variance errors.***

Regularization

Regularization helps to solve over fitting problem which implies model performing well on training data but performing poorly on validation (test) data. Regularization solves this problem by adding a penalty term to the objective function and control the model complexity using that penalty term.

Regularization is generally useful in the following situations:

1. Large number of variables
2. Low ratio of number observations to number of variables
3. High Multi-Collinearity

L1 Loss function or L1 Regularization
In L1 regularization we try to minimize the objective function by adding a penalty term to the sum of the absolute values of coefficients. This is also known as least absolute deviations method. Lasso Regression makes use of L1 regularization.

L2 Loss function or L2 Regularization
In L2 regularization we try to minimize the objective function by adding a penalty term to thesum of the squares of coefficients.RidgeRegression or shrinkage regression makes use of L2 regularization.

In the linear regression objective function we try to minimize the sum of squares of errors. In ridge regression (also known as shrinkage regression) we add a constraint on the sum of squares of the regression coefficients. Thus in ridge regression our objective function is:

$$\text{Min } (\sum \varepsilon^2 + \lambda \sum \beta^2) = \text{Min } \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_k X_k))^2 + \lambda \sum \beta^2$$

Here λ is the regularization parameter which is a non negative number. Here we do not assume normality in the error terms.
Very Important Note: We do not regularize the intercept term. The constraint is just on the sum of squares of regression coefficients of X's. We can see that ridge regression makes use of L2 regularization.

On solving the above objective function we can get the estimates of β as:

$$\hat{\beta} = (X'X + \lambda I)^{-1} X'y$$

Where I is an identity matrix of suitable order.

How can we choose the regularization parameter λ?

If we choose lambda = 0 then we get back to the usual OLS estimates. If lambda is chosen to be very large then it will lead to underfitting. Thus it is highly important to determine a desirable value of lambda.To tackle this issue, we plot the parameter estimates against different values of lambda and select the minimum value of λ after which the parameters tend to stabilize.

Lasso Regression

Lasso stands for Least Absolute Shrinkage and Selection Operator. It makes use of L1 regularization technique in the objective function. Thus the objective function in LASSO regression becomes:

$$\text{Min } (\sum \varepsilon^2 + \lambda \sum |\beta|) = \text{Min } \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_k X_k))^2 + \lambda \sum |\beta|$$

λ is the regularization parameter and the intercept term is not regularized. We do not assume that the error terms are normally distributed. For the estimates we don't have any specific mathematical formula but we can obtain the estimates using some statistical software. Note that lasso regression also needs standardization.

Advantage of lasso over ridge regression: Lasso regression can perform in-built variable selection as well as parameter shrinkage. While using ridge regression one may end up getting all the variables but with Shrinked Paramaters.

Which one is better - Ridge regression or Lasso regression?
Both ridge regression and lasso regression are addressed to deal with multicollinearity. Ridge regression is computationally more efficient over lasso regression.Any of them can perform better. So the best approach is to select that regression model which fits the test set data well.

Elastic Net Regression

Elastic Net regression is preferred over both ridge and lasso regression when one is dealing with highly correlated independent variables. It is a combination of both L1 and L2 regularization. The objective function in case of Elastic Net Regression is:

$$\text{Min } (\sum \varepsilon^2 + \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|) = \text{Min } \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_k X_k))^2 \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|$$

Like ridge and lasso regression, it does not assume normality.