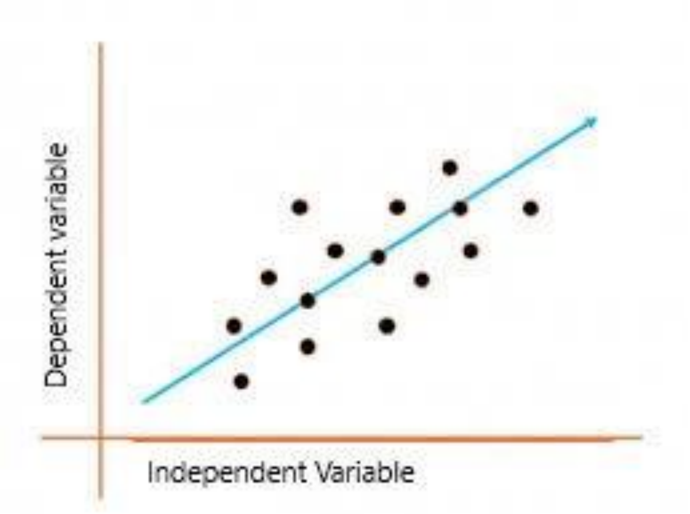
Linear Regression Notes

Simple Linear Regression

Linear regression is a quiet and the simplest statistical regression method used for predictive analysis in machine learning. Linear regression shows the linear relationship between the independent(predictor) variable i.e. X-axis and the dependent(output) variable i.e. Y-axis, called linear regression. If there is a single input variable **X**(dependent variable), such linear regression is called ***simple linear regression***.



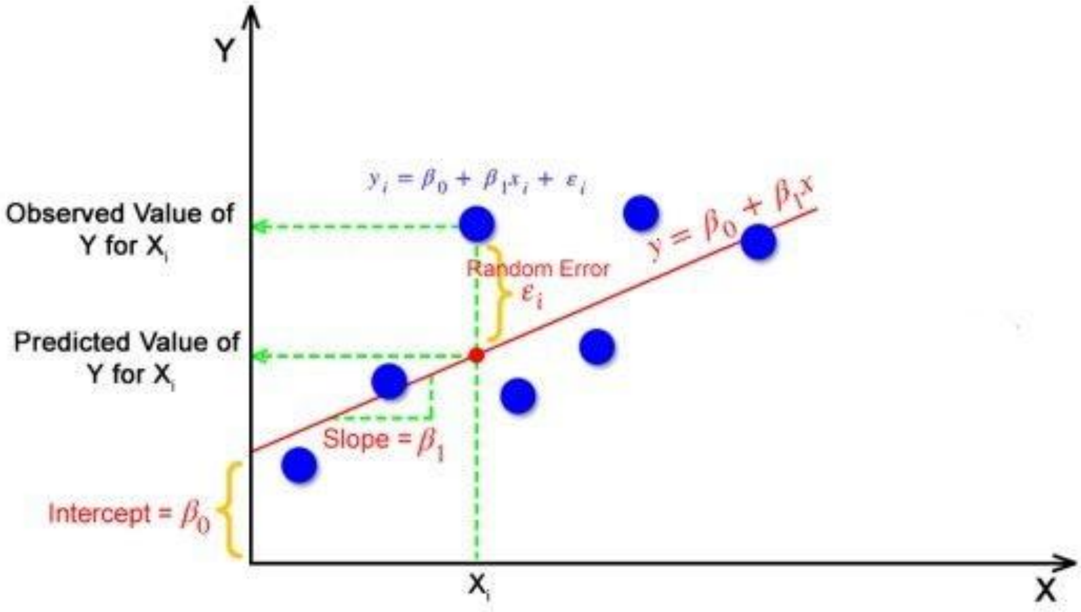
The above graph presents the linear relationship between the output(y) variable and predictor(X) variables. The blue line is referred to as the *best fit* straight line. Based on the given data points, we attempt to plot a line that fits the points the best.

To calculate best-fit line linear regression uses a traditional slope-intercept form which is given below,

$\mathbf{Y_i = \beta_0 + \beta_1 X_i}$

where Y_i = Dependent variable, **β_0** = constant/Intercept, **β_1** = Slope/Intercept, **X_i** = Independent variable.

This algorithm explains the linear relationship between the dependent(output) variable y and the independent(predictor) variable X using a straight line $Y= B_0 + B_1 X$.


Y
$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
Observed Value of Y for X_i
Random Error ε_i
$y = \beta_0 + \beta_1 x$
Predicted Value of Y for X_i
Slope = β_1
Intercept = β_0
X_i
X

But how the linear regression finds out which is the best fit line?

The goal of the linear regression algorithm is to get the **best values for B_0 and B_1** to find the best fit line. The best fit line is a line that has the least error which means the error between predicted values and actual values should be minimum.

Random Error(Residuals)

In regression, the difference between the observed value of the dependent variable(**y_i**) and the predicted value(**predicted**) is called the residuals.

$$\boldsymbol{\varepsilon_i = y_{predicted} - y_i}$$

where $y_{predicted} = B_0 + B_1 X_i$

-

What is the best fit line?

In simple terms, the best fit line is a line that fits the given scatter plot in the best way. Mathematically, the best fit line is obtained by minimizing the Residual Sum of Squares(RSS).

Cost Function for Linear Regression

The cost function helps to work out the optimal values for B_0 and B_1, which provides the best fit line for the data points.

In Linear Regression, generally **Mean Squared Error (MSE)** cost function is used, which is the average of squared error that occurred between the $\mathbf{y}_{\mathbf{predicted}}$ and $\mathbf{y}_{\mathbf{i}}$.

We calculate MSE using simple linear equation y=mx+b:

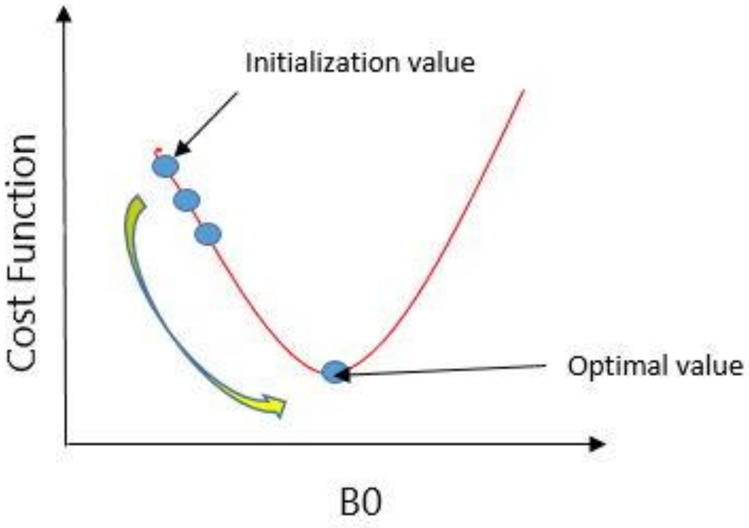
$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (\mathrm{B1}x_i + \mathrm{B0}))^2$$

Using the MSE function, we'll update the values of B_0 and B_1 such that the MSE value settles at the minima. These parameters can be determined using the gradient descent method such that the value for the cost function is minimum.

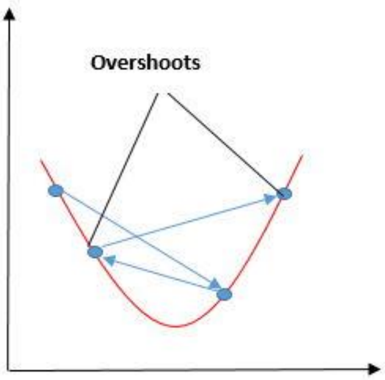
Gradient Descent for Linear Regression

Gradient Descent is one of the optimization algorithms that optimize the cost function(objective function) to reach the optimal minimal solution. To find the optimum solution we need to reduce the cost function(MSE) for all data points. This is done by updating the values of B_0 and B_1 iteratively until we get an optimal solution.

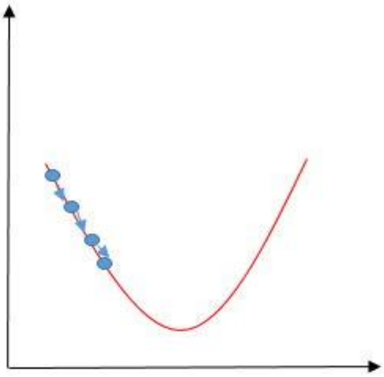
A regression model optimizes the gradient descent algorithm to update the coefficients of the line by reducing the cost function by randomly selecting coefficient values and then iteratively updating the values to reach the minimum cost function.



Let's take an example to understand this. Imagine a U-shaped pit. And you are standing at the uppermost point in the pit, and your motive is to reach the bottom of the pit. Suppose there is a treasure at the bottom of the pit, and you can only take a discrete number of steps to reach the bottom. If you opted to take one step at a time, you would get to the bottom of the pit in the end but, this would take a longer time. If you decide to take larger steps each time, you may achieve the bottom sooner but, there's a probability that you could overshoot the bottom of the pit and not even near the bottom. In the gradient descent algorithm, the number of steps you're taking can be considered as the **learning rate**, and this decides how fast the algorithm **converges** to the minima.



High learning rate



Low learning rate

To update B_0 and B_1, we take gradients from the cost function. To find these gradients, we take partial derivatives for B_0 and B_1.

$$J = \frac{1}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial B_0} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial B_1} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i) \cdot x_i$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial B_0} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i)$$

$$\frac{\partial J}{\partial B_1} = \frac{2}{n}\sum_{i=1}^{n}(B_0 + B_1 \cdot x_i - y_i) \cdot x_i$$

$$B_0 = B_0 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$B_1 = B_1 - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

We need to minimize the cost function J. One of the ways to achieve this is to apply the batch gradient descent algorithm. In batch gradient descent, the values are updated in each iteration. (Last two equations shows the updating of values)

The partial derivates are the gradients, and they are used to update the values of B_0 and B_1. Alpha is the learning rate.

Evaluation Metrics for Linear Regression

The strength of any linear regression model can be assessed using various evaluation metrics. These evaluation metrics usually provide a measure of how well the observed outputs are being generated by the model.

The most used metrics are,

1. Coefficient of Determination or R-Squared (R2)
2. Root Mean Squared Error (RSME) and Residual Standard Error (RSE)

1. Coefficient of Determination or R-Squared (R2)

R-Squared is a number that explains the amount of variation that is explained/captured by the developed model. It always ranges between 0 & 1 . Overall, the higher the value of R-squared, the better the model fits the data.

Mathematically it can be represented as,

$$\mathbf{R^2 = 1 - (RSS/TSS)}$$

- **Residual sum of Squares (RSS)** is defined as the sum of squares of the residual for each data point in the plot/data. It is the measure of the difference between the expected and the actual observed output.

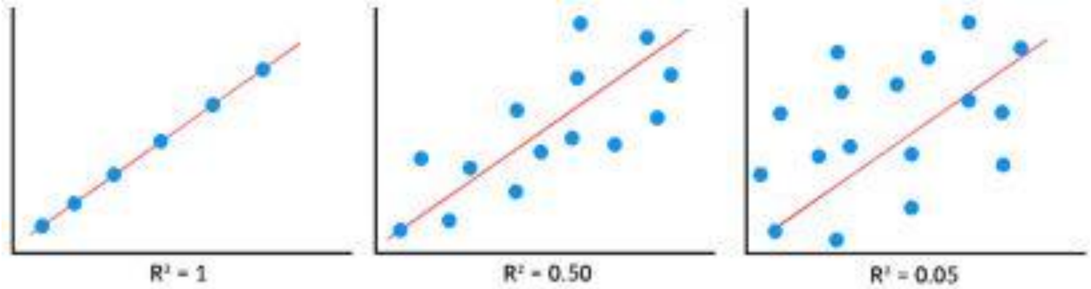
$$\mathrm{RSS} = \sum_{i=1}^{n}(y_i - b_0 - b_1 x_i)^2$$

- **Total Sum of Squares (TSS)** is defined as the sum of errors of the data points from the mean of the response variable. Mathematically TSS is,

$$TSS = \sum (y_i - \bar{y_i})^2$$

where y hat is the mean of the sample data points.

The significance of R-squared is shown by the following figures,



2. Root Mean Squared Error

The Root Mean Squared Error is the square root of the variance of the residuals. It specifies the absolute fit of the model to the data i.e. how close the observed data points are to the predicted values. Mathematically it can be represented as,

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\sum_{i=1}^{n}\left(y_i^{Actual} - y_i^{Predicted}\right)^2 \Big/ n}$$

To make this estimate unbiased, one has to divide the sum of the squared residuals by the **degrees of freedom** rather than the total number of data points in the model. This term is then called the **Residual Standard Error(RSE)**. Mathematically it can be represented as,

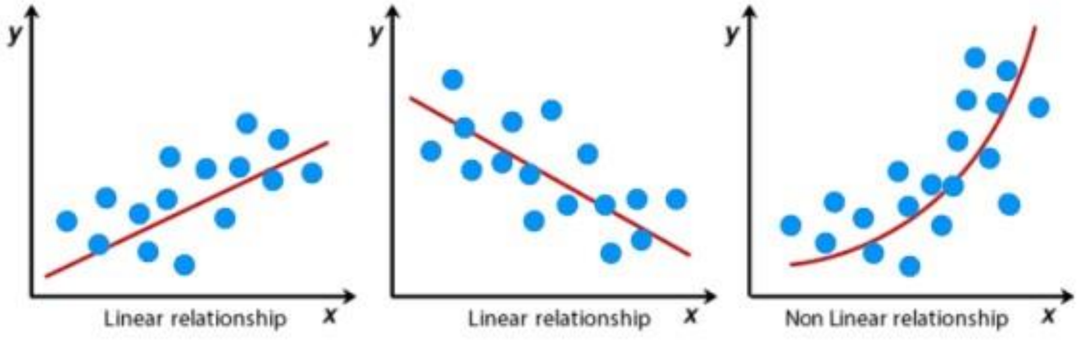
$$RSE = \sqrt{\frac{RSS}{df}} = \sqrt{\sum_{i=1}^{n}\left(y_i^{Actual} - y_i^{Predicted}\right)^2 \Big/ (n-2)}$$

R-squared is a better measure than RSME. Because the value of Root Mean Squared Error depends on the units of the variables (i.e. it is not a normalized measure), it can change with the change in the unit of the variables.

Assumptions of Linear Regression

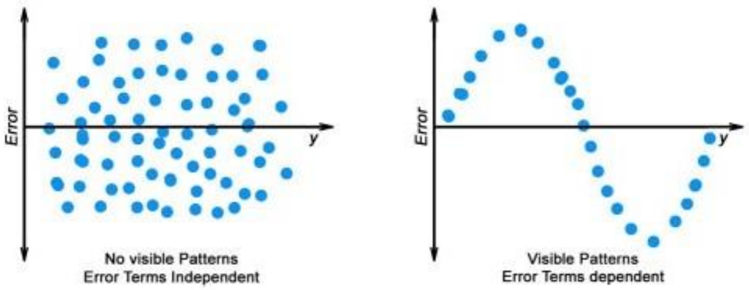
Regression is a parametric approach, which means that it makes assumptions about the data for the purpose of analysis. For successful regression analysis, it's essential to validate the following assumptions.

1. **Linearity of residuals:** There needs to be a linear relationship between the dependent variable and independent variable(s).

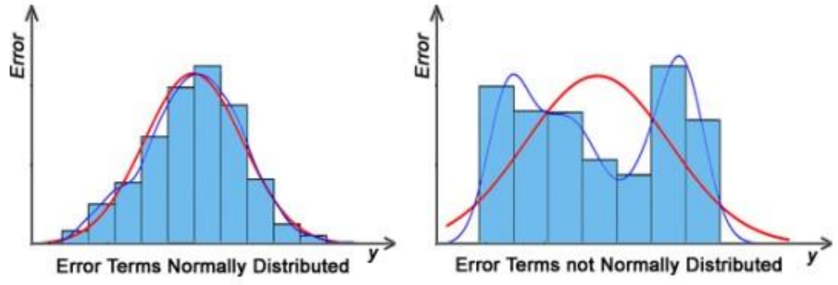


2. **Independence of residuals:** The error terms should not be dependent on one another (like in time-series data wherein the next value is dependent on the previous one). There should be no correlation between the residual terms. The absence of this phenomenon is known as **Autocorrelation.**

There should not be any visible patterns in the error terms.

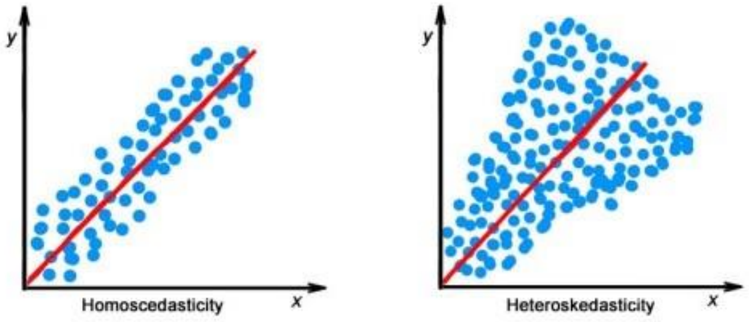


3. <u>Normal distribution of residuals:</u> The mean of residuals should follow a normal distribution with a mean equal to zero or close to zero. This is done in order to check whether the selected line is actually the line of best fit or not. If the error terms are non-normally distributed, suggests that there are a few unusual data points that must be studied closely to make a better model.



<u>4. The equal variance of residuals:</u> The error terms must have constant variance. This phenomenon is known as **Homoscedasticity**.

The presence of non-constant variance in the error terms is referred to as **Heteroscedasticity**. Generally, non-constant variance arises in the presence of *outliers or extreme leverage values*.



Hypothesis in Linear Regression

Once you have fitted a straight line on the data, you need to ask, "**Is this straight line a significant fit for the data?**" Or "I**s the beta coefficient explain the variance in the data plotted?"** And here comes the idea of **hypothesis testing** on the beta coefficient. The Null and Alternate hypotheses in this case are:

H_0: $B_1 = 0$

H_A: $B_1 \neq 0$

To test this hypothesis we use a **t-test,** test statistics for the beta coefficient is given by

$$t = \frac{m - \mu}{s/\sqrt{n}}$$

Assessing the model fit
Some other parameters to assess a model are:

1. **t statistic:** It is used to determine the p-value and hence, helps in determining whether the coefficient is significant or not
2. **F statistic**: It is used to assess whether the overall model fit is significant or not. Generally, the higher the value of the F-statistic, the more significant a model turns out to be.

Multiple Linear Regression

Multiple linear regression is a technique to understand the relationship between a *single* dependent variable and *multiple* independent variables.
The formulation for multiple linear regression is also similar to simple linear regression with
the small change that instead of having one beta variable, you will now have betas for all the variables used. The formula is given as:

$Y = B_0 + B_1X_1 + B_2X_2 + \ldots + B_pX_p + \boldsymbol{\varepsilon}$

Considerations of Multiple Linear Regression

All the four assumptions made for Simple Linear Regression still hold true for Multiple Linear Regression along with a few new additional assumptions.

1. **Overfitting**: When more and more variables are added to a model, the model may become far too complex and usually ends up memorizing all the data points in the training set. This phenomenon is known as the overfitting of a model. This usually leads to high training accuracy and very low test accuracy.

2. **Multicollinearity**: It is the phenomenon where a model with several independent variables, may have some variables interrelated.
3. **Feature Selection:** With more variables present, selecting the optimal set of predictors from the pool of given features (many of which might be redundant) becomes an important task for building a relevant and better model.

Multicollinearity

As multicollinearity makes it difficult to find out which variable is actually contributing towards the prediction of the response variable, it leads one to conclude incorrectly, the effects of a variable on the target variable. Though it does not affect the precision of the predictions, it is essential to properly detect and deal with the multicollinearity present in the model, as random removal of any of these correlated variables from the model causes the coefficient values to swing wildly and even change signs.
Multicollinearity can be detected using the following methods.

1. **Pairwise Correlations:** Checking the pairwise correlations between different pairs of independent variables can throw useful insights in detecting multicollinearity.
2. **Variance Inflation Factor (VIF):** Pairwise correlations may not always be useful as it is possible that just one variable might not be able to completely explain some other variable but some of the variables combined could be ready to do this. Thus, to check these sorts of relations between variables, one can use VIF. VIF basically explains the relationship of one independent variable with all the other independent variables. VIF is given by,

$$\mathrm{VIF} = \frac{1}{1-R^2}$$

where i refers to the i^{th} variable which is being represented as a linear combination of the rest of the independent variables.

The common heuristic followed for the VIF values is if VIF > 10 then the value is definitely high and it should be dropped. And if the VIF=5 then it may be valid but should be inspected first. If VIF < 5, then it is considered a good vif value.

Overfitting and Underfitting in Linear Regression

There have always been situations where a model performs well on training data but not on the test data. While training models on a dataset, overfitting, and underfitting are the most common problems faced by people.

Before understanding overfitting and underfitting one must know about bias and variance.

Bias:

Bias is a measure to determine how accurate is the model likely to be on future unseen data. Complex models, assuming there is enough training data available, can do predictions accurately. Whereas the models that are too naive, are very likely to perform badly with respect to predictions. Simply, Bias is errors made by training data.

Generally, linear algorithms have a high bias which makes them fast to learn and easier to understand but in general, are less flexible. Implying lower predictive performance on complex problems that fail to meet the expected outcomes.

Variance:

Variance is the sensitivity of the model towards training data, that is it quantifies how much the model will react when input data is changed.

Ideally, the model shouldn’t change too much from one training dataset to the next training data, which will mean that the algorithm is good at picking out the hidden underlying patterns between the inputs and the output variables.

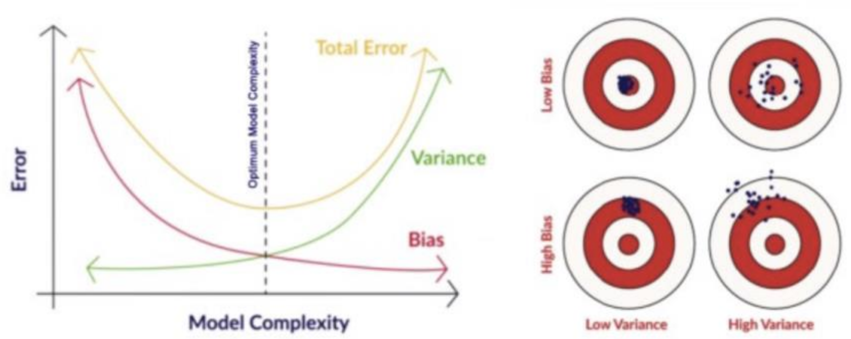
Ideally, a model should have lower variance which means that the model doesn’t change drastically after changing the training data(it is generalizable). Having higher variance will make a model change drastically even on a small change in the training dataset.

Let’s understand what is a bias-variance trade-off is.

Bias Variance Trade-off

The aim of any supervised machine learning algorithm is to achieve low bias and low variance as it is more robust. So that the algorithm should achieve better performance.

There is no escape from the relationship between bias and variance in machine learning.



There is an inverse relationship between bias and variance,

- An increase in bias will decrease the variance.
- An increase in the variance will decrease the bias.

There is a trade-off that plays between these two concepts and the algorithms must find a balance between bias and variance.

As a matter of fact, one cannot calculate the real bias and variance error terms because we do not know the actual underlying target function.

Now coming to the overfitting and underfitting.

Overfitting

When a model learns each and every pattern and noise in the data to such extent that it affects the performance of the model on the unseen future dataset, it is referred to as ***overfitting***. The model fits the data so well that it interprets noise as patterns in the data.

When a model has low bias and higher variance it ends up memorizing the data and causing overfitting. Overfitting causes the model to become specific rather than generic. This usually leads to high training accuracy and very low test accuracy.

Detecting overfitting is useful, but it doesn't solve the actual problem. There are several ways to prevent overfitting, which are stated below:

- Cross-validation
- If the training data is too small to train add more relevant and clean data.
- If the training data is too large, do some feature selection and remove unnecessary features.
- Regularization

Underfitting:

Underfitting is not often discussed as often as overfitting is discussed. When the model fails to learn from the training dataset and is also not able to generalize the test dataset, is referred to as ***underfitting***. This type of problem can be very easily detected by the performance metrics.

When a model has high bias and low variance it ends up not generalizing the data and causing underfitting. It is unable to find the hidden underlying patterns from the data. This usually leads to low training accuracy and very low test accuracy. The ways to prevent underfitting are stated below,

- Increase the model complexity
- Increase the number of features in the training data
- Remove noise from the data.

Source: link