Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Underfitting refers to a model that can neither model the training data nor generalize to new data. An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data. Underfitting is often not discussed as it is easy to detect given a good performance metric. The remedy is to move on and try alternate machine learning algorithms. Nevertheless, it does provide a good contrast to the problem of overfitting.

the model can become biased if

- you use the same data to train it as you do to test it

- your data is not representative of real world data

Important to test validation of method

* Report the accuracy of the model
* Should be understood
* Should be able to infer the same performance on new unseen data
* Should have scientific grounds or merits for the performance analysis

The training data set in Machine Learning is the actual dataset used to train the model for performing various actions. This is the actual data the ongoing development process models learn with various API and algorithm to train the machine to work automatically

Test set: a set of examples used only to assess the performance of a fully-trained classifier In the MLP case, we would use the test to estimate the error rate after we have chosen the final model (MLP size and actual weights) After assessing the final model on the test set, YOU MUST NOT tune the model any further!

percentage split is where you take your dataset and split it into a testing set and training set

[Stratified sampling](https://en.wikipedia.org/wiki/Stratified_sampling) aims at splitting one data set so that each split are similar with respect to something.

In a classification setting, it is often chosen to ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set.

As a result, if the data set has a large amount of each class, stratified sampling is pretty much the same as random sampling. But if one class isn't much represented in the data set, which may be the case in your dataset since you plan to oversample the minority class, then stratified sampling may yield a different target class distribution in the train and test sets than what random sampling may yield.

Note that the stratified sampling may also be designed to equally distribute some features the next train and test set. For example, if each sample represent one individual, and one feature is age, it is sometimes useful to have the same age distribution in both the train and test set.

K fold cross Validation

As there is never enough data to train your model, removing a part of it for validation poses a problem of underfitting. By reducing the training data, we risk losing important patterns/ trends in data set, which in turn increases error induced by bias. So, what we require is a method that provides ample data for training the model and also leaves ample data for validation. K Fold cross validation does exactly that.

In K Fold cross validation, the data is divided into k subsets. Now the holdout method is repeated k times, such that each time, one of the k subsets is used as the test set/ validation set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get total effectiveness of our model. As can be seen, every data point gets to be in a validation set exactly once, and gets to be in a training set k-1 times. This significantly reduces bias as we are using most of the data for fitting, and also significantly reduces variance as most of the data is also being used in validation set. Interchanging the training and test sets also adds to the effectiveness of this method. As a general rule and empirical evidence, K = 5 or 10 is generally preferred, but nothing’s fixed and it can take any value.

If the number of values belonging to each class are unbalanced, using stratified sampling is a good thing. You are basically asking the model to take the training and test set such that the class proportion is same as of the whole dataset, which is the right thing to do. If your classes are balanced then a shuffle (no stratification needed here) can basically guarantee a fair test and train split.

Now your model will be capable or at least enough equipped to predict the outnumbered class (class with lesser points in number). That is why instead of just calculating Accuracy, you have been given other metrics like Sensitivity and Specificity. Keep a watch on these, these are the guardians.

Supplied test – useful for To examine real world predictions even if no classification is known, Also useful if you want to create a stratified training and test set

Confusion Matrix Machine Learning

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix.

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

Definition of the Terms:

• Positive (P) : Observation is positive (for example: is an apple).

• Negative (N) : Observation is not positive (for example: is not an apple).

• True Positive (TP) : Observation is positive, and is predicted to be positive.

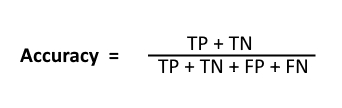
• False Negative (FN) : Observation is positive, but is predicted negative.

• True Negative (TN) : Observation is negative, and is predicted to be negative.

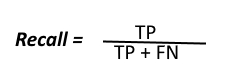
• False Positive (FP) : Observation is negative, but is predicted positive.

Accuracy:

Accuracy is given by the relation:



Sensitivity:  
Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN). Sensitivity is given by the relation:



**Classification vs Regression**

Classification predictive modelling problems are different from regression predictive modelling problems.

Classification is the task of predicting a discrete class label.

Regression is the task of predicting a continuous quantity.

There is some overlap between the algorithms for classification and regression; for example:

A classification algorithm may predict a continuous value, but the continuous value is in the form of a probability for a class label.

A regression algorithm may predict a discrete value, but the discrete value in the form of an integer quantity.

Some algorithms can be used for both classification and regression with small modifications, such as decision trees and artificial neural networks. Some algorithms cannot, or cannot easily be used for both problem types, such as linear regression for regression predictive modeling and logistic regression for classification predictive modeling.

Importantly, the way that we evaluate classification and regression predictions varies and does not overlap, for example:

Classification predictions can be evaluated using accuracy, whereas regression predictions cannot.

Regression predictions can be evaluated using root mean squared error, whereas classification predictions cannot.

Root-mean-squared error:

RMSE is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

What Is Sensitivity

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Sensitivity is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease. In other words, the person who is unhealthy actually got predicted as unhealthy.

Mathematically, sensitivity can be calculated as the following:

Sensitivity = (True Positive)/(True Positive + False Negative)

The following is the details in relation to True Positive and False Negative used in the above equation.

True Positive = Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.

False Negative = Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.

The higher value of sensitivity would mean higher value of true positive and lower value of false negative. The lower value of sensitivity would mean lower value of true positive and higher value of false negative. For healthcare and financial domain, models with high sensitivity will be desired.

What Is Specificity?

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate. The sum of specificity and false positive rate would always be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease. In other words, the person who is healthy actually got predicted as healthy is specificity.

Mathematically, specificity can be calculated as the following:

Specificity = (True Negative)/(True Negative + False Positive)

The following is the details in relation to True Negative and False Positive used in the above equation.

True Negative = Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true negative represents the number of persons who are healthy and are predicted as healthy.

False Positive = Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.

The higher value of specificity would mean higher value of true negative and lower false positive rate. The lower value of specificity would mean lower value of true negative and higher value of false positive.

What Are the Differences Between Sensitivity and Specificity?

While Sensitivity measure is used to determine the proportion of actual positive cases, which got predicted correctly, Specificity measure is used to determine the proportion of actual negative cases, which got predicted correctly.

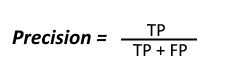
Where Are Sensitivity and Specificity Used?

Sensitivity and Specificity measures are used to plot the ROC curve. And, Area under ROC curve (AUC) is used to determine the model performance. The following represents different ROC curves and related AOC values.

Precision:

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP).

Precision is given by the relation:



F score:

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.

