**Cross-Validation**

Cross-Validation is a resampling technique that helps to make our model sure about its efficiency and accuracy on the unseen data. It is a method for evaluating Machine Learning models by training several other Machine learning models on subsets of the available input data set and evaluating them on the subset of the data set.

We have different types of Cross-Validation techniques but let’s see the basic functionality of Cross-Validation: The first step is to divide the cleaned data set into K partitions of equal size.

1. Then we need to treat the Fold-1 as a test fold while the other K-1 as train folds and compute the score of the test-fold.
2. We need to repeat step 2 for all folds taking another fold as a test while remaining as a train.
3. Last step would be to take the average of scores of all the folds.

**Types of Cross-Validation**

**1. Holdout Method**

This technique works on removing a part of the training data set and sending that to a model that was trained on the rest of the data set to get the predictions. We then calculate the error estimation which tells how our model is doing on unseen data sets. This is known as the **Holdout Method.**

**Pros**

1. This Method is Fully independent of data.
2. This Method only needs to be run once so has lower computational costs.

**Cons**

1. The Performance is subject to higher variance given the smaller size of the data.

**2. K-Fold Cross-Validation**

In a Data-Driven World, there is never enough data to train your model, on top of that removing a part of it for validation poses a greater problem of Underfitting and we risk losing important patterns and trends in our data set, which in turn increases Bias. So ideally, we require a method that provides ample amounts of data for training the model and leaves ample amounts of data for validation sets.

In K-Fold cross-validation, the data is divided into k subsets or we can take it as a holdout method repeated k times, such that each time, one of the k subsets is used as the validation set and the other k-1 subsets as the training set. The error is averaged over all k trials to get the total efficiency of our model.

We can see that each data point will be in a validation set exactly once and will be in a training set k-1 time. This helps us reduce bias as we are using most of the data for fitting and reduces variance as most of the data is also being used in the validation set.

**Pros**

1. This will help to overcome the problem of computational power.
2. Models may not be affected much if an outlier is present in data.
3. It helps us overcome the problem of variability.

**Cons**

1. Imbalanced data sets will impact our model.

**3. Stratified K-Fold Cross-Validation**

K Fold Cross Validation technique will not work as expected for an Imbalanced Data set. When we have an imbalanced data set, we need a slight change to the K Fold cross validation technique, such that each fold contains approximately the same strata of samples of each output class as the complete. This variation of using a stratum in K Fold Cross Validation is known as Stratified K Fold Cross Validation.

**Pros**

1. It can improve different models using hyper-parameter tuning.
2. Helps us compare models.
3. It helps in reducing both Bias and Variance.

**4. Leave-P-Out Cross-Validation**

In this approach we leave p data points out of training data out of a total n data points, then n-p samples are used to train the model and p points are used as the validation set. This is repeated for all combinations, and then the error is averaged.

**Pros**

1. It has Zero randomness
2. The Bias will be lower

**Cons**

1. This method is exhaustive and computationally infeasible.