Cross Validation

In order to evaluate the performance of the model. we divide our data into training data and test data.

Now, the training and test sets are randomly selected and may give us different results over another different selection of train, test data.

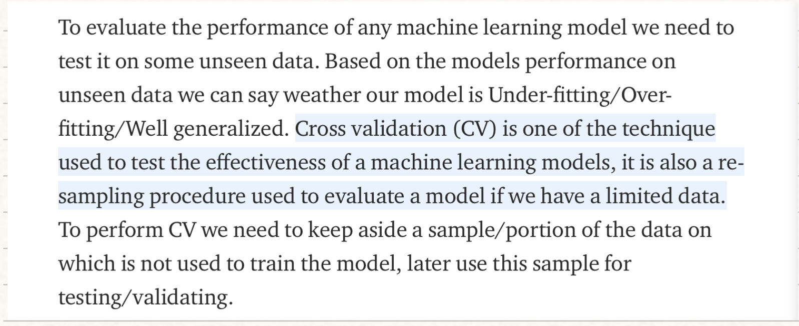
Accuracy of data may fluctuate when we use different random state of data.

1000 records random_state = 1

75% training 25% testing → 85% accuracy

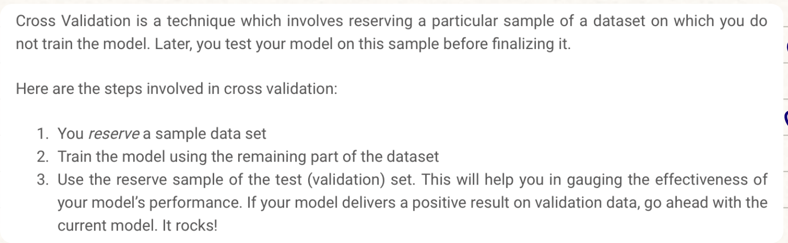
1000 records random_state = 10

75% training 25% testing → 87% accuracy



To evaluate the performance of any machine learning model we need to test it on some unseen data. Based on the models performance on unseen data we can say weather our model is Under-fitting/Over-fitting/Well generalized. Cross validation (CV) is one of the technique used to test the effectiveness of a machine learning models, it is also a re-sampling procedure used to evaluate a model if we have a limited data. To perform CV we need to keep aside a sample/portion of the data on which is not used to train the model, later use this sample for testing/validating.

Cross validation is a technique in which we train our model using the subset of the dataset and then evaluate using complementary subset of the dataset. Steps:-

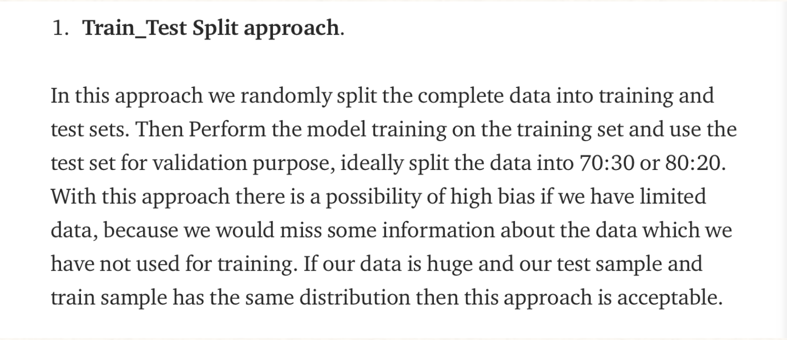


Cross Validation is a technique which involves reserving a particular sample of a dataset on which you do not train the model. Later, you test your model on this sample before finalizing it.

Here are the steps involved in cross validation:

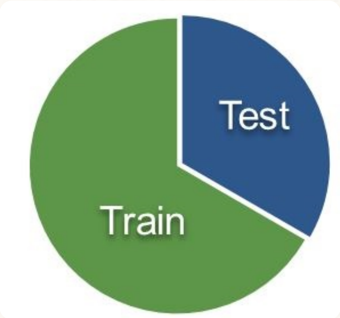
1. You *reserve* a sample data set
2. Train the model using the remaining part of the dataset
3. Use the reserve sample of the test (validation) set. This will help you in gauging the effectiveness of your model's performance. If your model delivers a positive result on validation data, go ahead with the current model. It rocks!

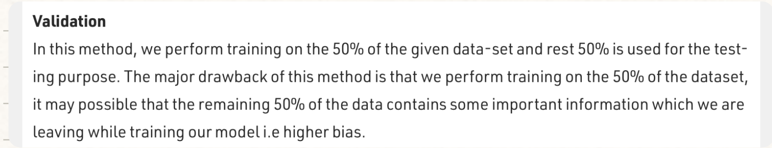
(i) Reserve some portion of sample data-set.
(ii) Using the rest data-set, train the model.
(iii) Test the model using the reserve portion of the data-set.



1. **Train_Test Split approach**.

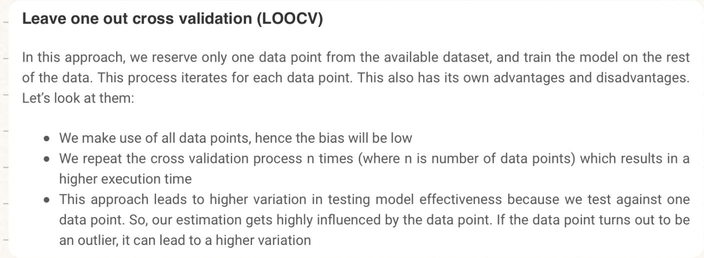
In this approach we randomly split the complete data into training and test sets. Then Perform the model training on the training set and use the test set for validation purpose, ideally split the data into 70:30 or 80:20. With this approach there is a possibility of high bias if we have limited data, because we would miss some information about the data which we have not used for training. If our data is huge and our test sample and train sample has the same distribution then this approach is acceptable.





Validation

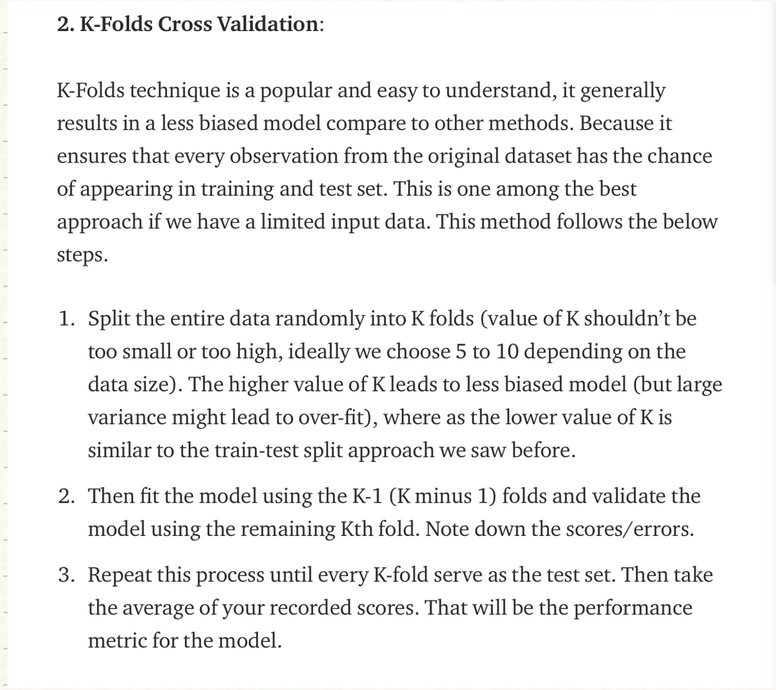
In this method, we perform training on the 50% of the given data-set and rest 50% is used for the testing purpose. The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model i.e higher bias.



Leave one out cross validation (LOOCV)

In this approach, we reserve only one data point from the available dataset, and train the model on the rest of the data. This process iterates for each data point. This also has its own advantages and disadvantages. Let's look at them:

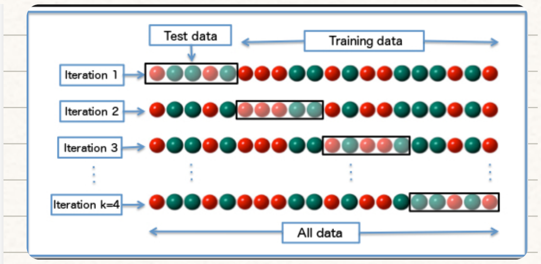
- We make use of all data points, hence the bias will be low
- We repeat the cross validation process n times (where n is number of data points) which results in a higher execution time
- This approach leads to higher variation in testing model effectiveness because we test against one data point. So, our estimation gets highly influenced by the data point. If the data point turns out to be an outlier, it can lead to a higher variation

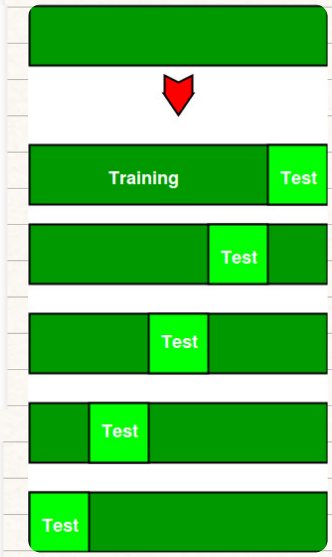


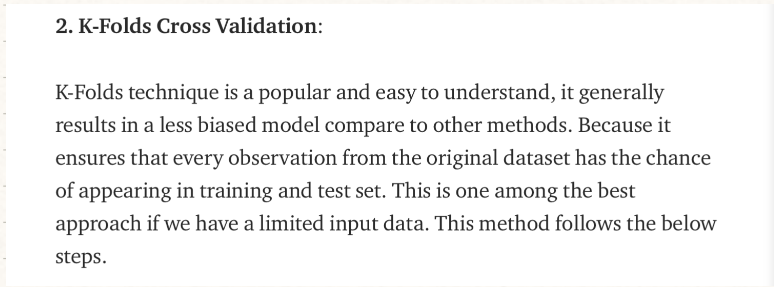
2. K-Folds Cross Validation:

K-Folds technique is a popular and easy to understand, it generally results in a less biased model compare to other methods. Because it ensures that every observation from the original dataset has the chance of appearing in training and test set. This is one among the best approach if we have a limited input data. This method follows the below steps.

1. Split the entire data randomly into K folds (value of K shouldn't be too small or too high, ideally we choose 5 to 10 depending on the data size). The higher value of K leads to less biased model (but large variance might lead to over-fit), where as the lower value of K is similar to the train-test split approach we saw before.
2. Then fit the model using the K-1 (K minus 1) folds and validate the model using the remaining Kth fold. Note down the scores/errors.
3. Repeat this process until every K-fold serve as the test set. Then take the average of your recorded scores. That will be the performance metric for the model.







2. K-Folds Cross Validation:

K-Folds technique is a popular and easy to understand, it generally results in a less biased model compare to other methods. Because it ensures that every observation from the original dataset has the chance of appearing in training and test set. This is one among the best approach if we have a limited input data. This method follows the below steps.