**CROSS VALIDATION**

*Ref –* **Analytics steps** ( [7 Types of Cross-Validation in Machine Learning | Analytics Steps](https://www.analyticssteps.com/blogs/7-types-cross-validation)), **Turing** ([Different Types of Cross-Validations in Machine Learning. (turing.com)](https://www.turing.com/kb/different-types-of-cross-validations-in-machine-learning-and-their-explanations))

A statistical concept that aims to evaluate a Machine Learning model and its predictive ability, Cross-Validation is a validation technique for assessing the functioning of a particular model with the help of samples obtained from a data set.

ML programming or AI libraries could help one develop machine learning models for prediction, cross-validation lets one conduct machine learning performance evaluation by sampling small samples obtained from the same data.

Cross-validation is a resampling method of evaluating the validity of an ML model using a data sample. A technique that lets one to weigh the overfitting or underfitting extent of a model using the training data and testing data, cross-validation also allows one to test the accuracy of a model before launching it for public use.

There are various types of cross-validation. However, mentioned above are the 7 most common types - Holdout, K-fold, Stratified k-fold, Rolling, Monte Carlo, Leave-p-out, and Leave-one-out method.

Although each one of these types has some drawbacks, they aim to test the accuracy of a model as much as possible. All in all, cross-validation is the only solution to identify the ability of a model to generalize independent data.

Once a model is all set for work, it is trained with the training data set to make it learn and work. Even when the machine has been trained, it is quite possible that it may present some challenges while being run.

Perhaps this is where cross-validation steps in. While there are numerous ways to cross-validate a model, all methods aim to test a data sample kept aside from the training data set.

A set of 3 steps are involved in the process of cross-validation. These are as follows -

1. Separate a part of the data set from the rest.
2. Utilize the rest of the data set to train the ML model.
3. Once the model is ready, validate the model using the data set that was separated earlier.

This separate data sample is treated as independent data that tests the ability of a model to generalize new data. Find out in the following segments the different types of cross-validations in order to understand how these methods work.

Why is it important?

While the primary significance of cross-validation is to validate a model and test its accuracy, there are more factors that make this method important. Let us find out.

More than validating the model and testing its accuracy, cross-validation is used for measuring overfitting and any other errors that might be noticed while testing the model. Overfitting refers to a concept wherein a statistical model works against its training data set and does not show accuracy as expected.

In this way, a model becomes overfitted and is unable to live up to its purpose. Similarly, cross-validation can also be used to identify and mitigate underfitting, a concept in ML that proves that a particular model can neither train as per the data set nor generalize independent data set.

For any model that is all set to work on independent data, it is important for it to be validated by cross-validation. Not only does this method suggest the level of accuracy the model has, but it also demarcates the level of deviation that the model has according to the standard expectation.

Simply put, cross-validation is the only solution to test the performance of a model before launching it.

**Types of Cross-Validation**

While there are 3 easy steps to conduct the method of cross-validation, there are numerous ways through which this process can be conducted. Herein, we will discover the most common types of cross-validations.

This banner introduces the readers to the different types of cross-validations.

**Holdout Cross-Validation**

The Holdout method is quite easy to understand and work upon. To get started, the data sample is divided into two parts (Training Data Set and Testing Data Set).

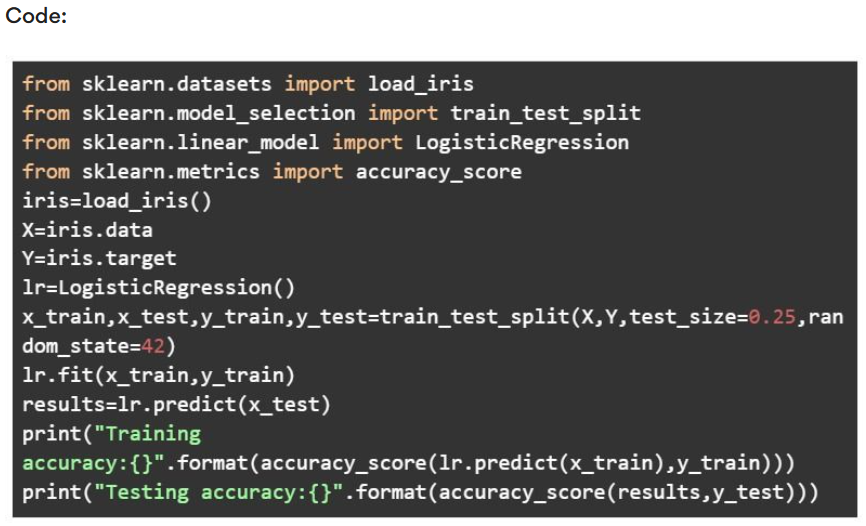
Before the division takes place, the data sample is shuffled so that samples get mixed and lead to an accurate training data set. As the training data set is twice the size of the test set in machine learning, the model is trained with a large number of samples as compared to the samples available in the testing data set.

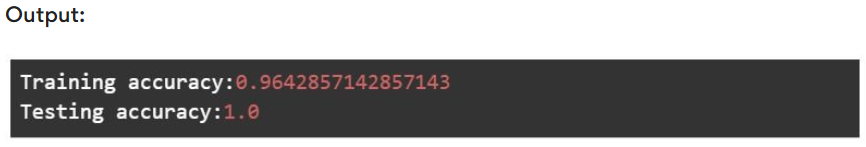
Usually, the ratio of training data set to testing data set is 70:30 or 80:20. The next step is to train the model with the training data set and once it is trained, the model is tested with the testing data set.

Although this method might seem to be easy and efficient, it has its own drawbacks.

While the training data set is kept to be more than the testing data set in terms of size, it could be possible that the training data set is not representative of the whole data sample.

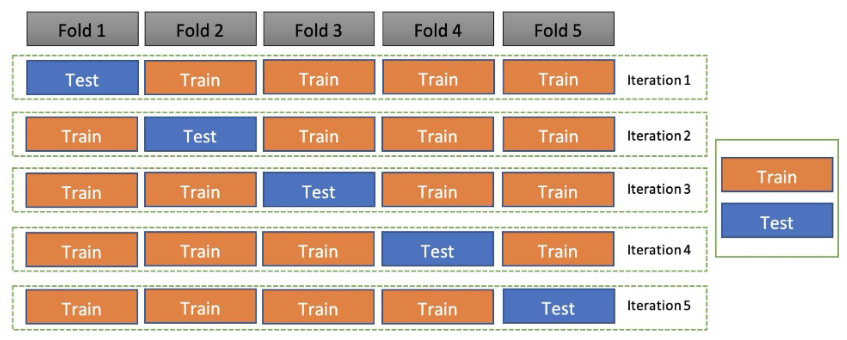
One of the disadvantages of the holdout method is that the testing data set could contain essential characteristics of the whole data that can get missed out. This method is also known as the train/test split approach.





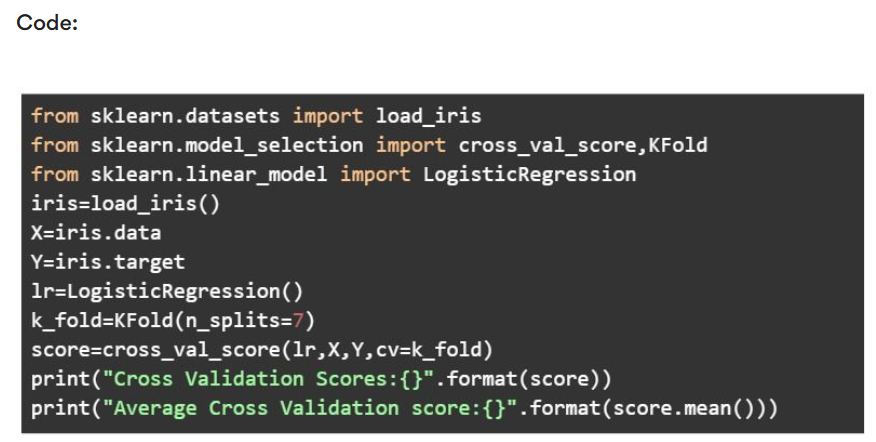
**K-fold Cross-Validation**

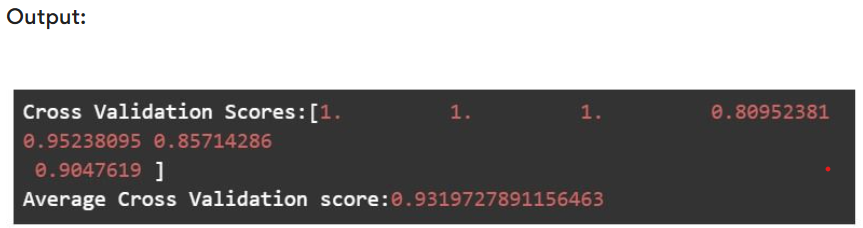
Another type of cross-validation is the K-fold cross-validation. The parameter for this type is 'K' which refers to the number of subsets or folds obtained from the data sample.



The first step is to train the model using the entire data set. The second step is to divide the data sample in 'k' number of subsets (let us suppose, 11). From here on, these subsets become the testing data sets that are then used for testing the validation of a model one by one.

This particular type of cross-validation is considered to be an unbiased and inclusive validation method as it involves the training and testing of almost every subset.



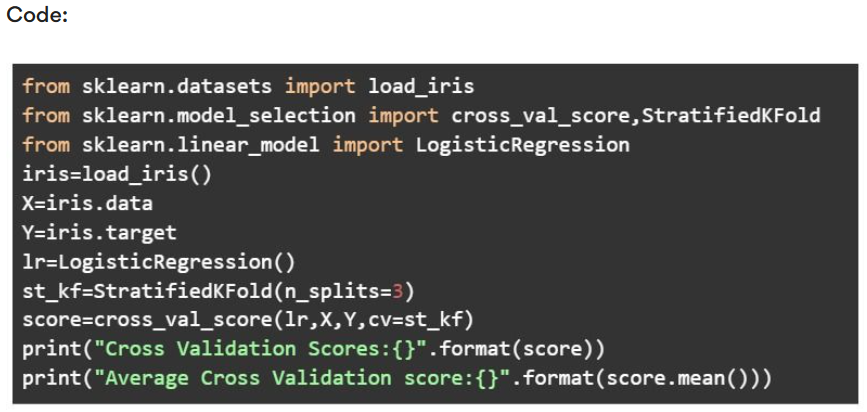


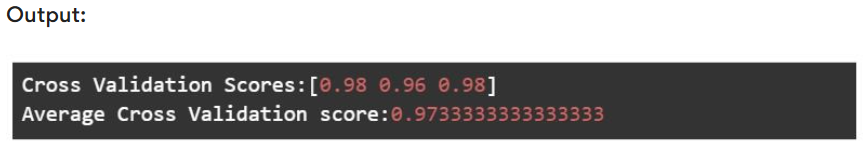
**Stratified K-fold Cross-Validation**

The stratified K-fold cross-validation method is yet another method that involves the division of data sample sets in 'k' subsets or folds.

However, in order to ensure that there is no biased division of data in 'k' folds, the process of stratification is conducted to rearrange the data in such a manner that each fold represents the whole data.

Especially in the case of classification in machine learning, sunsets must represent data from both classes or all classes. Perhaps stratification leads to the placement of data from all classes in all subsets such that the subsets wholly represent the data and lead to more accuracy.



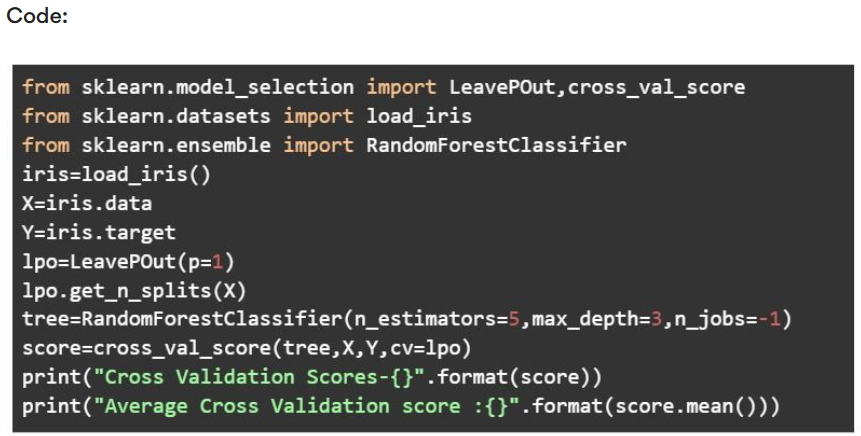


**Leave-p-out Cross-Validation**

Another type of cross-validation is the Leave-p-out cross-validation method. Herein, the data sample comprises data points (n). The total number of data points (n) is used to separate a set of data points that is used for testing.

These data points are referred to as (p). The training data set is obtained by calculating (n-p) and the model is trained accordingly. Once the training is done, p data points are used for cross-validation.

All possible combinations of p are tested on the model so as to get the maximum accuracy. As there are more than one accuracy level, an average is taken out to get the final accuracy of the model.



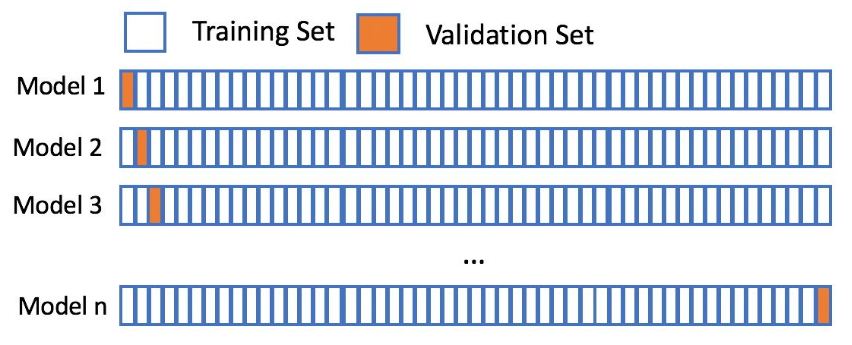


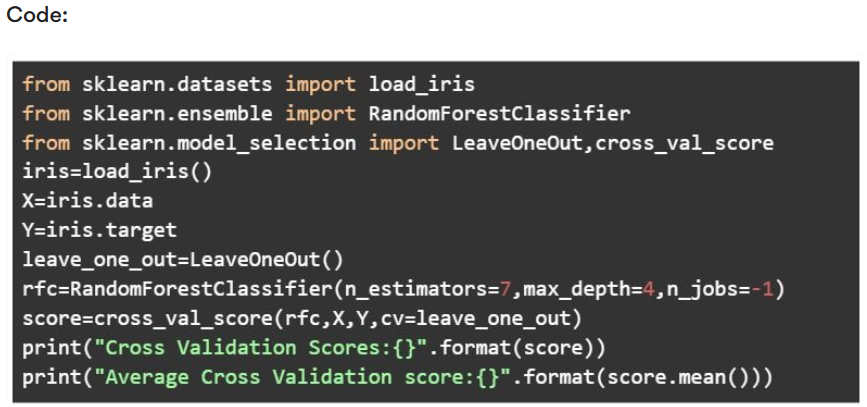
**Leave-one-out Cross-Validation**

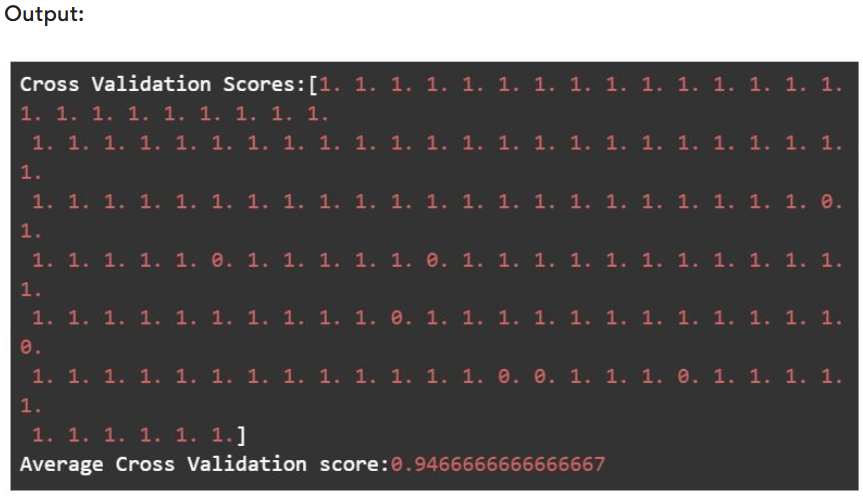
A variant of the Leave-p-out cross-validation method, the Leave-one-out cross-validation is another type of cross-validation. Herein, p is kept to be 1 (p=1) and the n-p data points are used to train the model.

Thus, when the training is done, the p data point or a single data point is used to validate the model.

One of the biggest drawbacks of this type is that a major part of the data sample is used for training the model, however, only a single data point is used to evaluate its accuracy. Therefore, this type is often considered to be an expensive method.

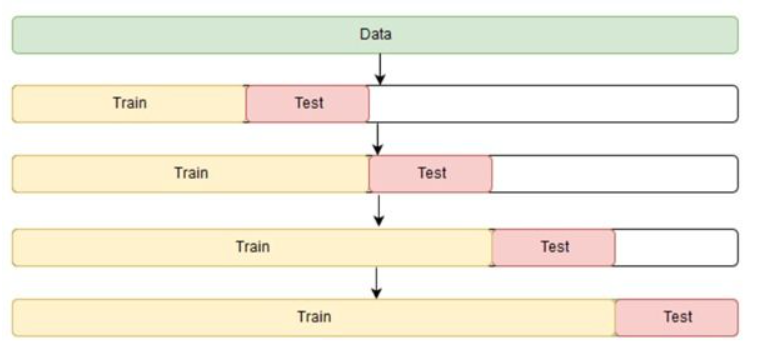






**Rolling Cross-Validation / Time Series Split Cross-Validation.**

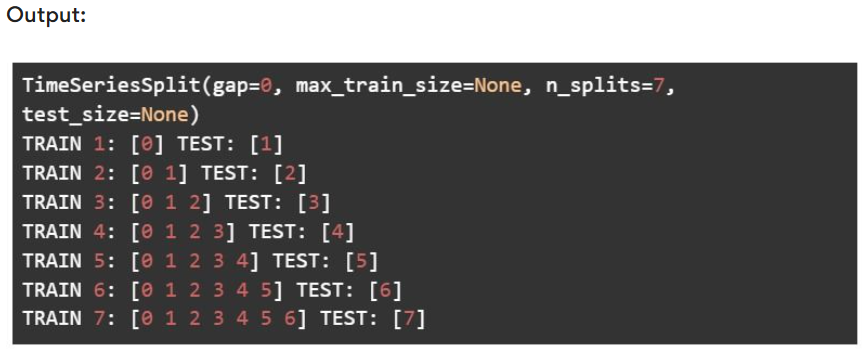
For data based on time series, no cross-validation method is effective except the rolling cross-validation method. This method involves taking a subset out of the data set that serves as the training data set.



A consequent subset is used for testing the data that helps go evaluate the accuracy of the model.

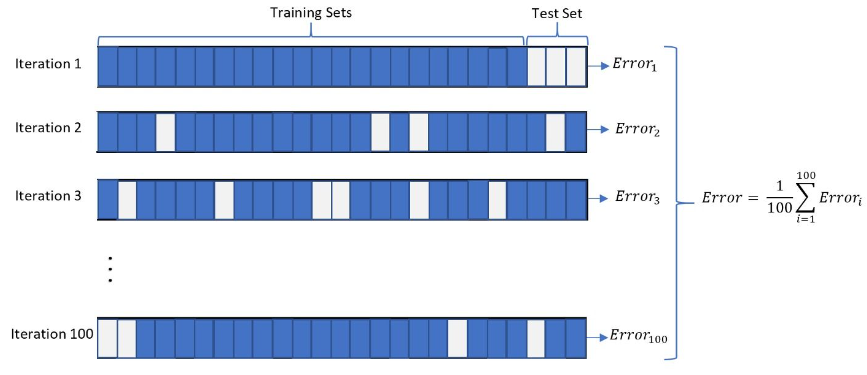
Once the testing is done, the process is repeated by taking a subset from the data sample, training the model with it, and testing another subset on the same. This is known as the rolling method of cross-validation as the subsets keep rolling until all data points are used for training and testing.





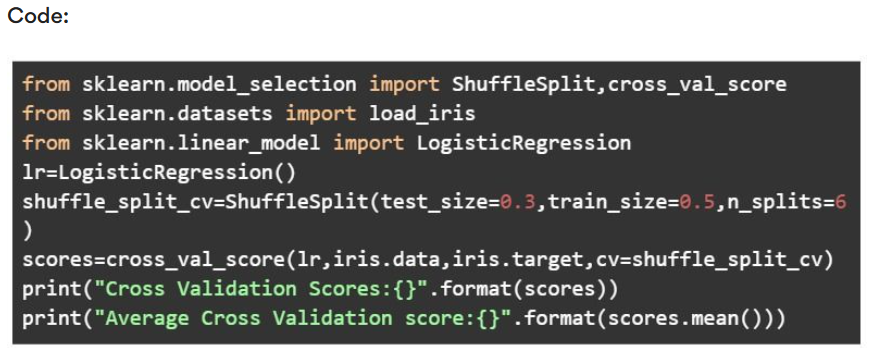
**Monte Carlo Cross-Validation**

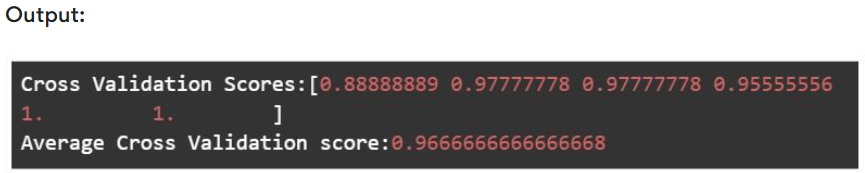
The Monte Carlo cross-validation method refers to a validation method that randomly picks a subset that is used to train the model. The remaining data points are used for testing the model.



This process is repeated a number of times as the same data sample is used for procuring training and testing data sets. Unlike the holdout method, the data sample is tried a number of times which ensures that accuracy is maintained and the model is validated in an unbiased manner.

One of the biggest disadvantages of this method is that particular data points could be left while some data points could be repeatedly used for training and testing.





**Conclusion**

To conclude, cross-validation is a resampling method of evaluating the validity of an ML model using a data sample. A technique that lets one to weigh the overfitting or underfitting extent of a model using the training data and testing data, cross-validation also allows one to test the accuracy of a model before launching it for public use.

There are various types of cross-validation. However, mentioned above are the 7 most common types - Holdout, K-fold, Stratified k-fold, Rolling, Monte Carlo, Leave-p-out, and Leave-one-out method.

Although each one of these types has some drawbacks, they aim to test the accuracy of a model as much as possible. All in all, cross-validation is the only solution to identify the ability of a model to generalize independent data.