Cross Validation in Machine Learning

- Cross-validation is a Statistical Method used to estimate the performance (or accuracy) of machine learning models.
- It is used to protect against overfitting in a predictive model, particularly in a case where the amount of data may be limited.
- In cross-validation, you make a fixed number of folds (or partitions) of the data, run the analysis on each fold, and then average the overall error estimate.

Types of Cross-Validation

1. K–Fold Cross Validation
2. Holdout cross-validation
3. Stratified k-fold cross-validation
4. Leave-p-out cross-validation
5. Leave-one-out cross-validation
6. Monte carlo(shuffle-split)
7. Time series(rolling cross-validation)

What is Cross-Validation?

Cross-validation is a technique for evaluating a machine learning model and testing its performance. It is a way to test how good a machine learning model is. It's like giving a test to see if the model can solve real-world problems. CV is commonly used in applied ML tasks.

The Core Algorithm of Cross-Validation

Cross-validation methods share a common algorithmic structure:

Data Split: The dataset is divided into two distinct subsets—one for training and the other for testing.

Model Training: The machine learning model is trained on the training dataset.

Model Validation: The trained model is then validated using the test dataset.

Repetitions: Steps 1 to 3 are repeated a number of times, which depends on the specific cross-validation technique employed.

1. **K-fold cross-validation**

- In this technique, the whole dataset is partitioned in k parts of equal size and each partition is called a fold.
- It's known as k-fold since there are k parts where k can be any integer - 3,4,5, etc.
- One fold is used for validation and other K-1 folds are used for training the model.
- To use every fold as a validation set and other left-outs as a training set, this technique is repeated k times until each fold is used once.

The algorithm of the k-Fold technique:

Divide into Folds: Split your data into 'k' equal parts, like dividing a pie into slices.

Test One Slice: Take one slice (fold) as a test set and the others as training sets.

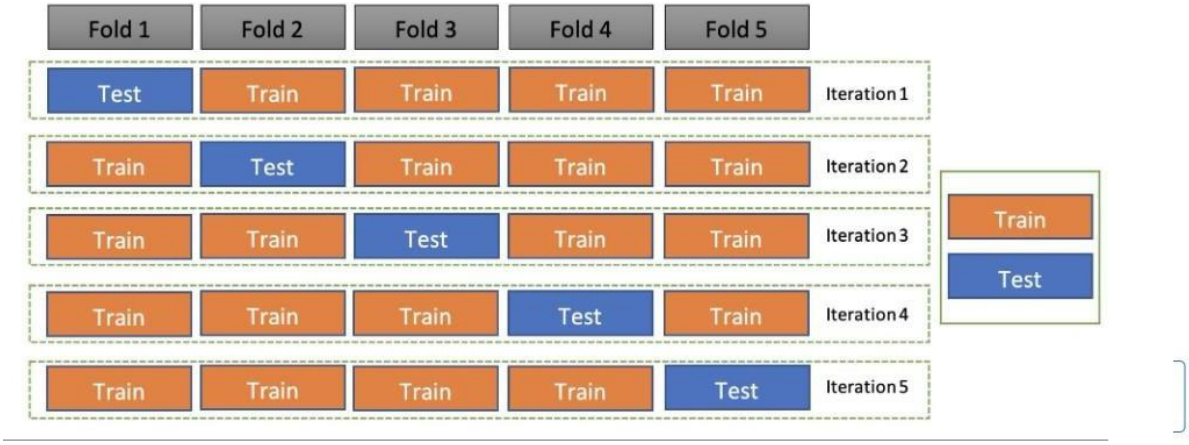
Train Model: Train a new model using the training slices.

Test Model: Test the model on the slice you set aside.

Repeat: Do this 'k' times, each time with a different slice as the test set.

Average Results: Average the results from all 'k' tests to see how well your model works overall.

In general, it is always better to use k-Fold technique instead of hold-out.



2. Holdout cross-validation

- It Also called a train-test split.
- holdout cross-validation has the entire dataset partitioned randomly into a training set and a validation set.
- A rule of thumb to partition data is that nearly 70% of the whole dataset will be used as a training set and the remaining 30% will be used as a validation set.
- Since the dataset is split into only two sets, the model is built just one time on the training set and executed faster.

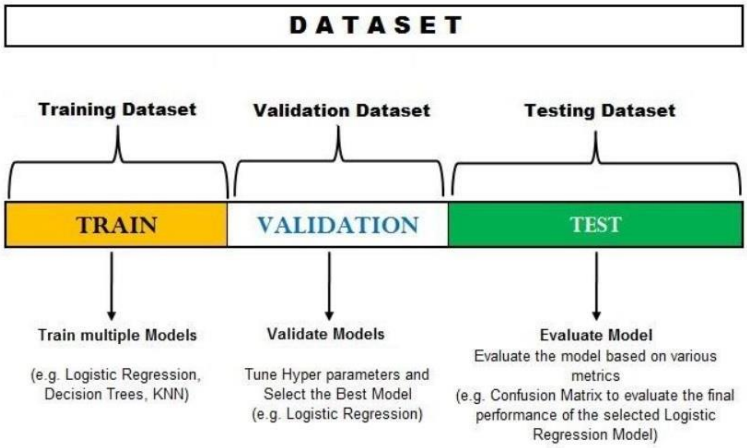
How It Works:

Data Split: You divide your dataset into two parts: the training set and the test set. Typically, 80% of the data goes into the training set, and 20% goes into the test set, but you can adjust these percentages as needed.

Model Training: You teach your model on the training set.

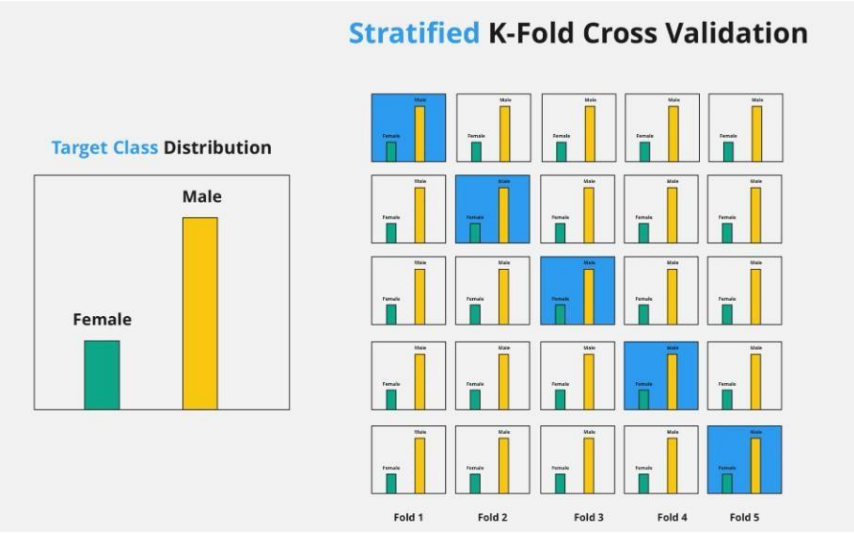
Model Testing: You test your model on the test set.

Result: You save the outcome of the test.



3. Stratified k-fold cross-validation

- k-fold validation can't be used for imbalanced datasets because data is split into k-folds with a uniform probability distribution.
- Not so with stratified k-fold, which is an enhanced version of the k-fold cross- validation technique.
- Although it too splits the dataset into k equal folds, each fold has the same ratio of instances of target variables that are in the complete dataset.
- This enables it to work perfectly for imbalanced datasets, but not for time- series data.



4.Leave-p-out cross-validation

- An exhaustive cross-validation technique, p samples are used as the validation set and n-p samples are used as the training set if a dataset has n samples.
- The process is repeated until the entire dataset containing n samples gets divided on the validation set of p samples and the training set of n-p samples.
- This continues till all samples are used as a validation set.
- The technique, which has a high computation time, produces good results.
- However, it's not considered ideal for an imbalanced dataset and is deemed to be a computationally unfeasible method.

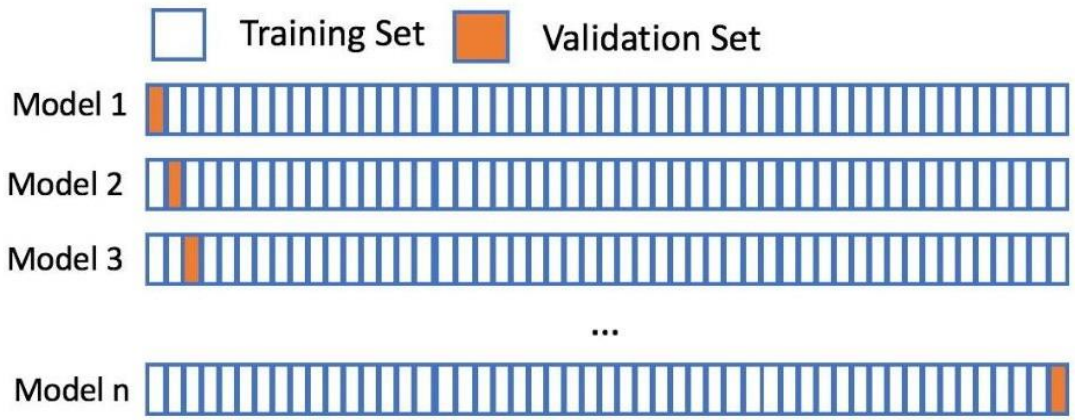
  This is because if the training set has all samples of one class, the model will not be able to properly generalize and will become biased to either of the classes.

5.Leave-one-out cross-validation

- In this technique, only 1 sample point is used as a validation set and the remaining n-1 samples are used in the training set.
- Think of it as a more specific case of the leave-p-out cross-validation technique with P=1.

To understand this better, consider this example:

There are 1000 instances in your dataset. In each iteration, 1 instance will be used for the validation set and the remaining 999 instances will be used as the training set. The process repeats itself until every instance from the dataset is used as a validation sample.



- The leave-one-out cross-validation method is computationally **expensive** to perform and **shouldn't be used with very large datasets**.
- The good news is that the technique is **very simple** and requires no configuration to specify.
- It also provides a **reliable** and **unbiased estimate** for your model performance.

6.Monte Carlo cross-validation

- It Also known as shuffle split cross-validation and repeated random subsampling cross-validation,
- the Monte Carlo technique involves splitting the whole data into training data and test data.
- Splitting can be done in the percentage of 70-30% or 60-40% - or anything you prefer.
- The only condition for each iteration is to keep the train-test split percentage different.
- The next step is to fit the model on the train data set in that iteration and calculate the accuracy of the fitted model on the test dataset.
- Repeat these iterations many times - 100,400,500 or even higher - and

take the average of all the test errors to conclude how well your model performs.

7.Time series (rolling cross-validation / forward chaining method)

- Time series is the type of data collected at different points in time.
- This kind of data allows one to understand what factors influence certain variables from period to period.
- this technique is used to perform cross-validation on time series data with time as the important factor.
- Some examples of time series data are weather records, economic indicators, etc.