**Methods to Evaluate and Cross-Validate Machine Learning Models**

**1. Train-Test Split**

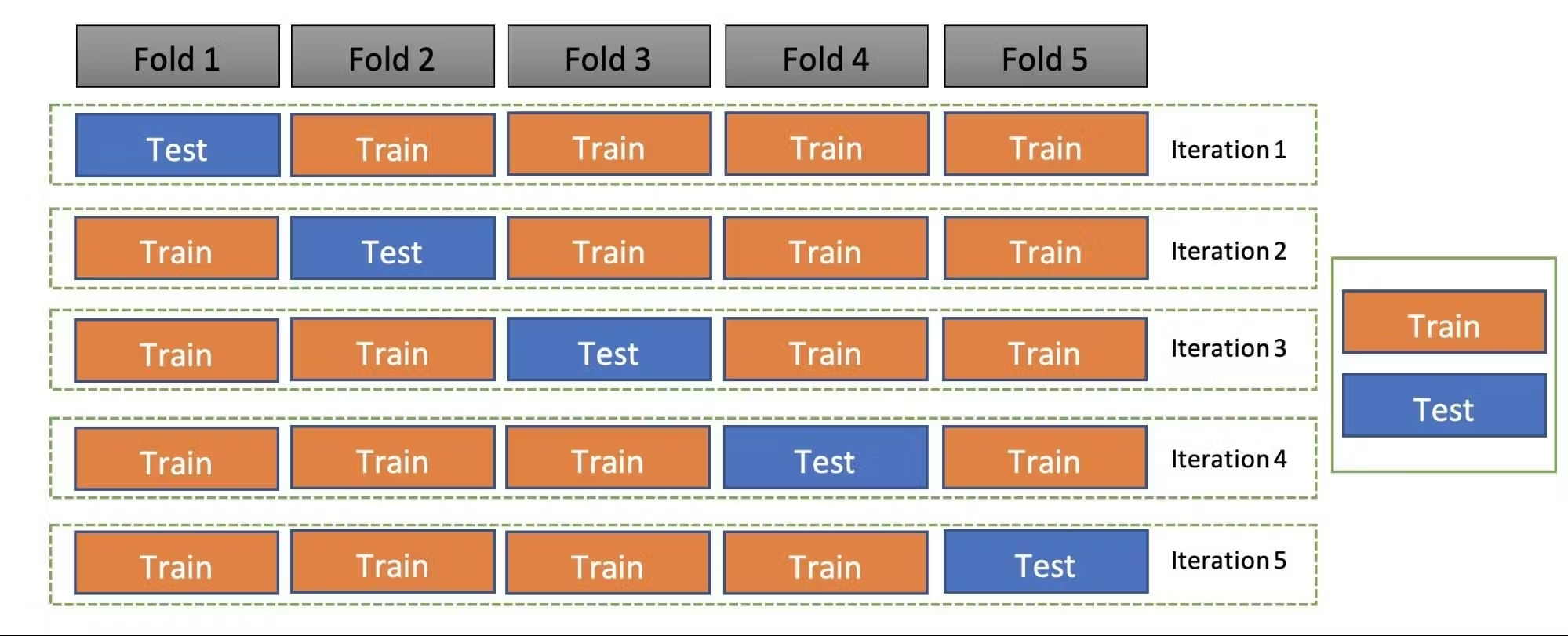
The simplest validation method where data is split into two subsets:

* Training set (70-80%) for model training
* Test set (20-30%) for final evaluation

2. **K-Fold Cross-Validation**

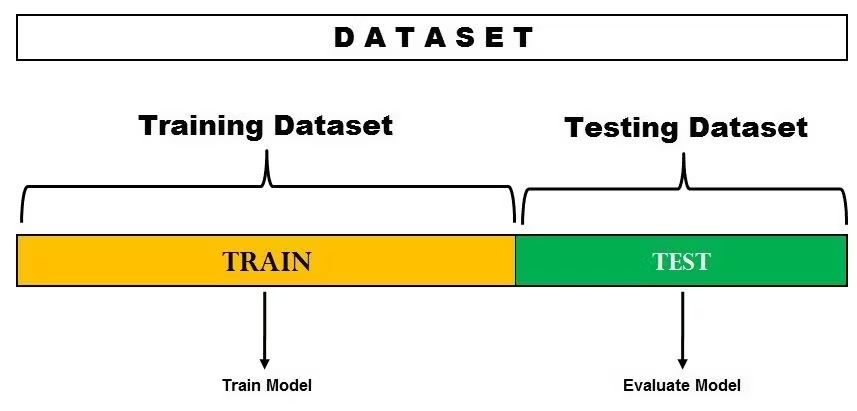
In this technique, the whole dataset is partitioned in k parts of equal size and each partition is called a fold. It’s known as k-fold since there are k parts where k can be any integer - 3,4,5, etc.

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.

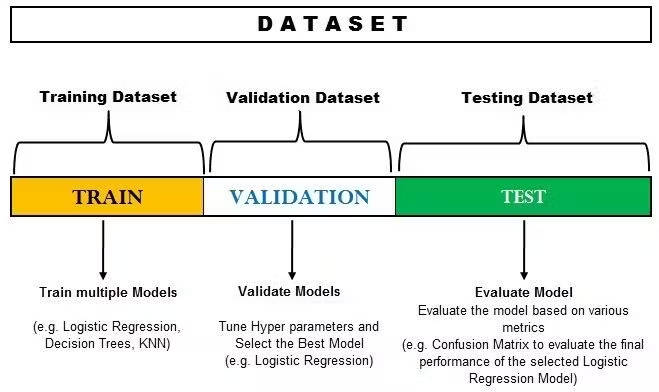


**3. Holdout cross-validation**

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.

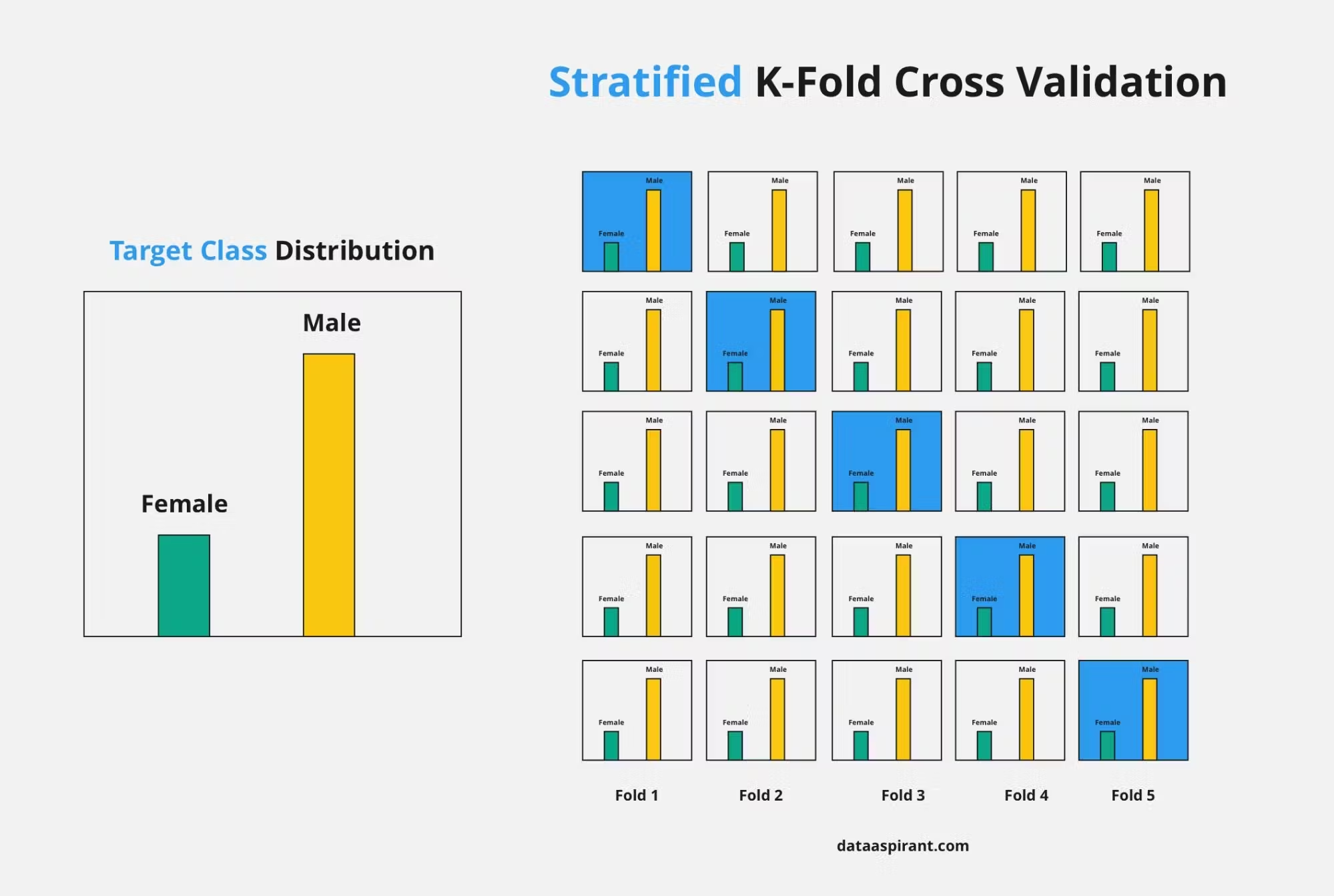


In the image above, the dataset is split into a training set and a test set. You can train the model on the training set and test it on the testing dataset. However, if you want to hyper-tune your parameters or want to select the best model, you can make a validation set like the one below.



**4. Stratified k-fold cross-validation**

As seen above, 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.



**5. Evaluation Metrics**

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

Confusion matrix is a table with four different combinations of predicted and actual values. It is extremely useful for measuring precision-recall, Specificity, Accuracy, and most importantly, AUC-ROC curves.

Here are a few definitions you need to remember for a confusion matrix:

* True Positive: You predicted positive, and it is true.
* True Negative: You predicted negative, and it is true.
* False Positive: (Type 1 Error): You predicted positive, and it is false.
* False Negative: (Type 2 Error): You predicted negative, and it is false.
* Accuracy: the proportion of the total number of correct predictions that were correct.
* Positive Predictive Value or Precision: the proportion of positive cases that were correctly identified.
* Negative Predictive Value: the proportion of negative cases that were correctly identified.
* Sensitivity or Recall: the proportion of actual positive cases which are correctly identified.
* Specificity: the proportion of actual negative cases which are correctly identified.

