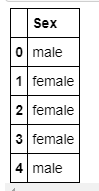
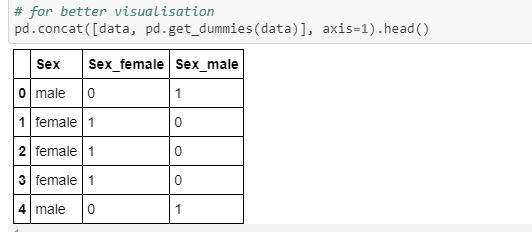
**One Hot Encoding**

* One hot encoding, consists of replacing the categorical variable by different boolean variables, which take **value 0 or 1**, to indicate whether or not a certain category / label of the variable was present for that observation.
* Each one of the boolean variables are also known as dummy variables or binary variables.

**Before One hot Encoding**



**After One Hot Encoding**



* One hot encoding is a process by which categorical variables are converted into a form that **could be provided to ML algorithms** to do a better job in prediction.
* For example, from the categorical variable "Gender", with labels 'female' and 'male', we can generate the boolean variable "female", which takes 1 if the person is female or 0 otherwise. We can also generate the variable male, which takes 1 if the person is "male" and 0 otherwise.
* As you may have noticed, we only need 1 of the 2 dummy variables to represent the original categorical variable Sex. Any of the 2 will suffice, and it doesn't matter which one we select, since they are equivalent.Therefore, to encode a categorical variable with 2 labels, we need 1 dummy variable.
* To extend this concept, to encode categorical variable with k labels, we need k-1 dummy variables.

**Advantages**

* Straightforward to implement
* Makes no assumption
* Keeps all the information of the categorical variable

**Disadvantages**

* Does not add any information that may make the variable more predictive
* If the variable has loads of categories, then OHE increases the feature space dramatically

**When should you use k and when k-1?**

* When the original variable is binary, that is, when the original variable has only 2 labels, then you should create one and only one binary variable.

***When the original variable has more than 2 labels, then following is important:***

**One hot encoding into k-1:**

* One hot encoding into k-1 binary variables takes into account that we can use 1 less dimension and still represent the whole information: if the observation is 0 in all the binary variables, then it must be 1 in the final (removed) binary variable.
* As an example, for the variable **gender encoded into male**, if the observation is 0, then it has to be female. We do not need the additional female variable to explain that.
* One hot encoding with k-1 binary variables should be used in linear regression, to keep the **correct number of degrees of freedom (k-1)**. The linear regression has access to all of the features as it is being trained, and therefore examines altogether the whole set of dummy variables.
* This means that k-1 binary variables give the whole information about (represent completely) the original categorical variable to the linear regression & the same is true for all machine learning algorithms that look at **ALL the features at the same time during training**. For example, support vector machines and neural networks as well. And clustering algorithms.

**One hot encoding into k dummy variables**

* However, tree based models select at each iteration only a group of features to make a decision. This is to separate the data at each node.
* Therefore, the last category, the one that was removed in the one hot encoding into k-1 variables, would only be taken into account by those splits or even trees, that use the entire set of binary variables at a time. And this would rarely happen, because each split usually uses 1-3 features to make a decision. So, tree based methods will never consider that additional label, the one that was dropped. Thus, if the categorical variables will be used in a tree based learning algorithm**, it is good practice to encode it into k binary variables instead of k-1.**
* Finally, if you are planning to do feature selection, you will also need the entire set of binary variables (k) to let the machine learning model select which ones have the most predictive power.

**Notes**

* If our datasets have a few multi-label variables, we will end up very soon with datasets with thousands of columns or more. And this may make training of our algorithms slow.
* In addition, many of these dummy variables may be similar to each other, since it is not unusual for 2 or more variables to share the same combinations of 1 and 0s.

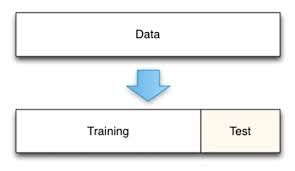
# **Types of Validation**

**Cross-Validation** also referred to **as out of sampling technique**. It is a resampling procedure used to evaluate machine learning models and access how the model will perform for an independent **test dataset**. In simple words it allows us to utilize our data even better.

Some of the different cross-validation techniques are:

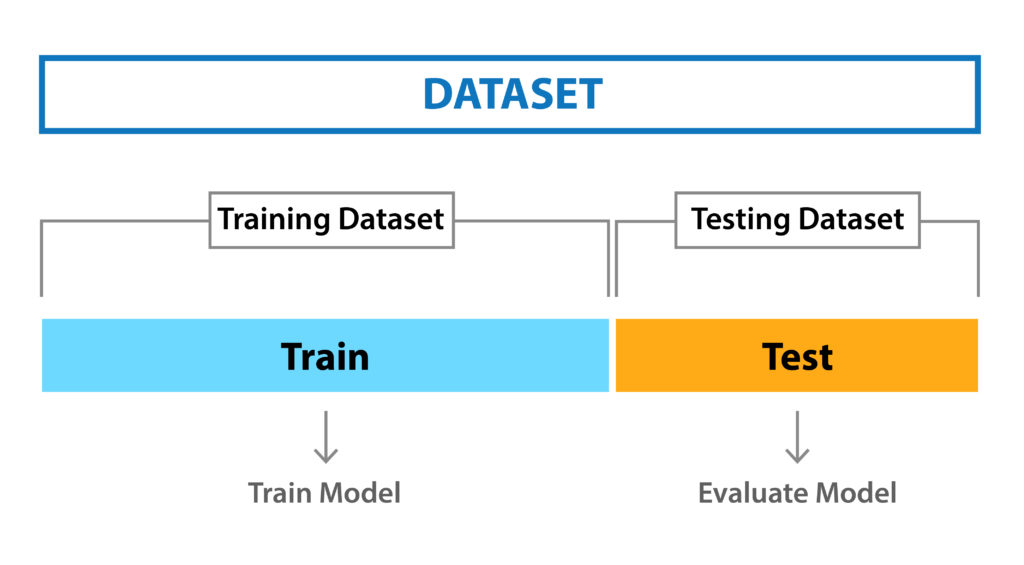
1. Holdout cross-validation
2. k-fold cross-validation
3. Stratified k-fold cross-validation
4. Leave one out cross-validation
5. Repeated random subsampling validation
6. Time Series cross-validation
7. Leave p out cross-validation

* We often randomly split the dataset into **train data** and **test data** to develop a machine learning model.
* The training data is used to **train the ML model** and the same model **is tested on independent testing data** to evaluate the performance of the model.
* With the change in the random state of the split, the accuracy of the model also changes, so we are not able to achieve a fixed accuracy for the model.



### **1.HoldOut Validation Approach- Train And Test Split:**

In this method, Dataset is **randomly split into train and test data**. split of training data is generally more than test data. The training data is used to induce the model and validation data is evaluates the performance of the model.



**Pros:** Simple, easy to understand, and implement.

**Cons:-**

1) Not suitable for an imbalanced dataset.

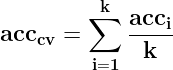
2)A lot of data is isolated from training set.

**2. k-fold cross-validation:**

In k-fold cross-validation, the original dataset is equally partitioned into k subparts or folds. Out of the k-folds or groups, for each iteration, one group is selected as validation data, and the remaining (k-1) groups are selected as training data.

The process is repeated for k times until each group is treated as validation and remaining as training data.

The final accuracy of the model is computed by taking the mean accuracy of the k-models validation data.





**Pros:**

1. The model has low bias.
2. Low time complexity
3. The entire dataset is utilized for both training and validation.

**Cons:**

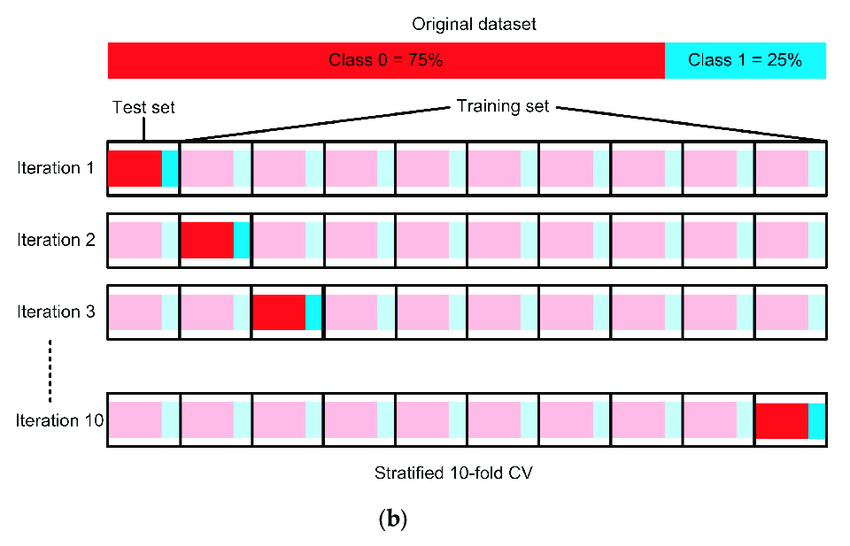
Not suitable for an imbalanced dataset.

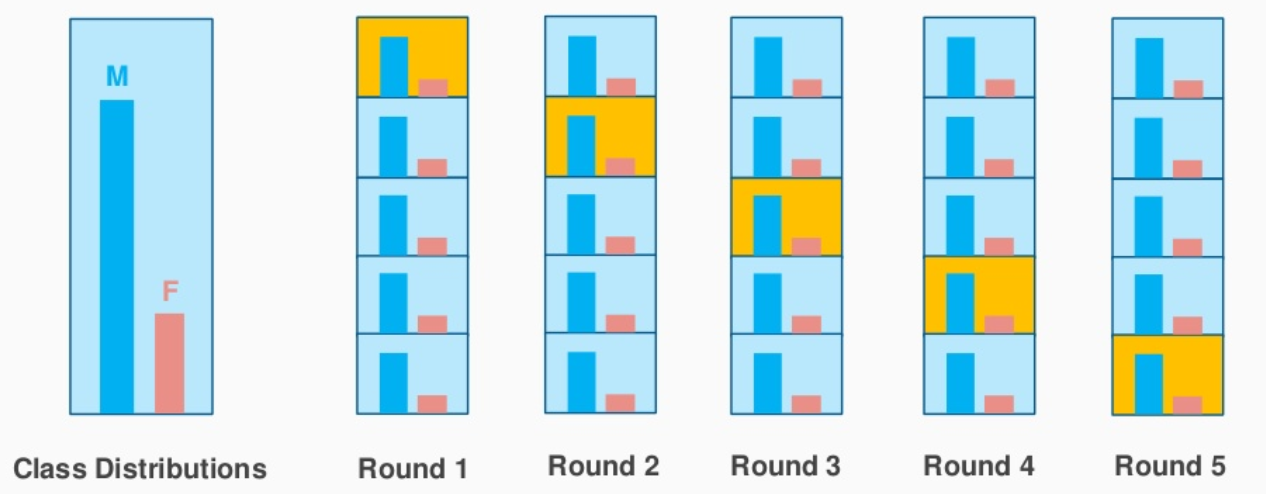
**3. Stratified k-fold cross-validation: -**

* Stratified k-fold cross-validation works well for an **imbalanced dataset**.
* In Stratified k-fold cross-validation, the dataset is partitioned into k groups or folds such that the validation data/Test set has an equal number of instances of target class label.
* This ensures that one particular class is not over present in the validation or train data especially when the dataset is imbalanced.
* The final score is computed by taking the mean of scores of each fold.

**Pros:** 1. Works well for an imbalanced dataset.

**Cons:** Now suitable for time series dataset.





**4. Leave-one-out cross-validation:**

* Leave-one-out cross-validation (LOOCV) is an exhaustive cross-validation technique. It is a category of LpOCV with the case of p=1.
* For a dataset having n rows, 1st row is selected for validation, and the rest (n-1) rows are used to train the model. For the next iteration, the 2nd row is selected for validation and rest to train the model. Similarly, the process is repeated until n steps or the desired number of operations.
* Exhaustive cross-validation methods are cross-validation methods that learn and test in all possible ways.

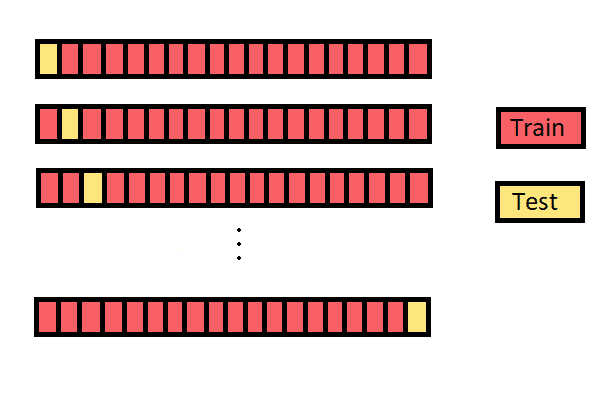
**Pros:**

Simple, easy to understand, and implement.

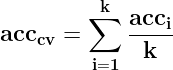
**Cons:**

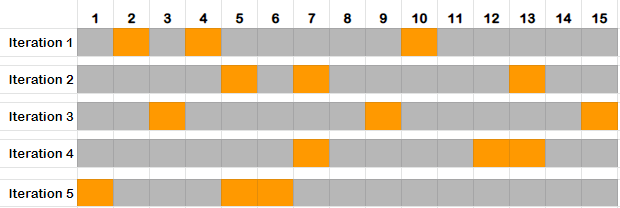
The model may lead to a low bias.

The computation time required is high.



* 1. **Repeated Random Test-Train Splits:-**
* Repeated random subsampling validation also referred to **as Monte Carlo cross-validation** splits the dataset randomly into training and validation. Unlikely k-fold cross-validation split of the dataset into not in groups or folds but splits in this case in random.
* The number of iterations is not fixed and decided by analysis. The results are then averaged over the splits.
* This technique is a hybrid of **traditional train-test splitting** and **the k-fold cross-validation** method. In this technique, we create random splits of the data in the training-test set manner and then repeat the process of splitting and evaluating the algorithm multiple times, just like the cross-validation method.





**Pros:**

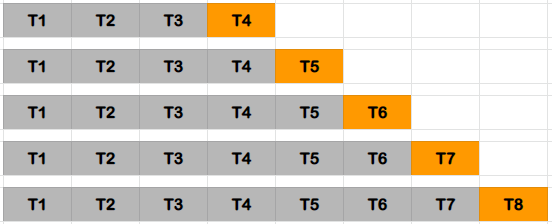
* The proportion of train and validation splits is not dependent on the number of iterations or partitions.

**Cons:**

* Some samples may not be selected for either training or validation.
* Not suitable for an imbalanced dataset.

**6. Time Series cross-validation:**

* The order of the data is very important for time-series related problem. For time-related dataset random split or k-fold split of data into train and validation may not yield good results.
* For the time-series dataset, the split of data into train and validation is according to the time also referred to as**forward chaining method**or **rolling cross-validation**. For a particular iteration, the next instance of train data can be treated as validation data.



As mentioned in the above diagram, for the 1st iteration, 1st 3 rows are considered as train data and the next instance T4 is validation data. The chance of choice of train and validation data is forwarded for further iterations.

**7) Leave p-out cross-validation:**

* Leave p-out cross-validation (LpOCV) is an exhaustive cross-validation technique, that involves using p-observation as validation data, and remaining data is used to train the model.
* This is repeated in all ways to cut the original sample on a validation set of p observations and a training set.
* A variant of LpOCV with p=2 known as leave-pair-out cross-validation has been recommended as a nearly unbiased method for estimating the area under ROC curve of a binary classifier.
* A particular case of this method is when p = 1. This is known as Leave one out cross validation