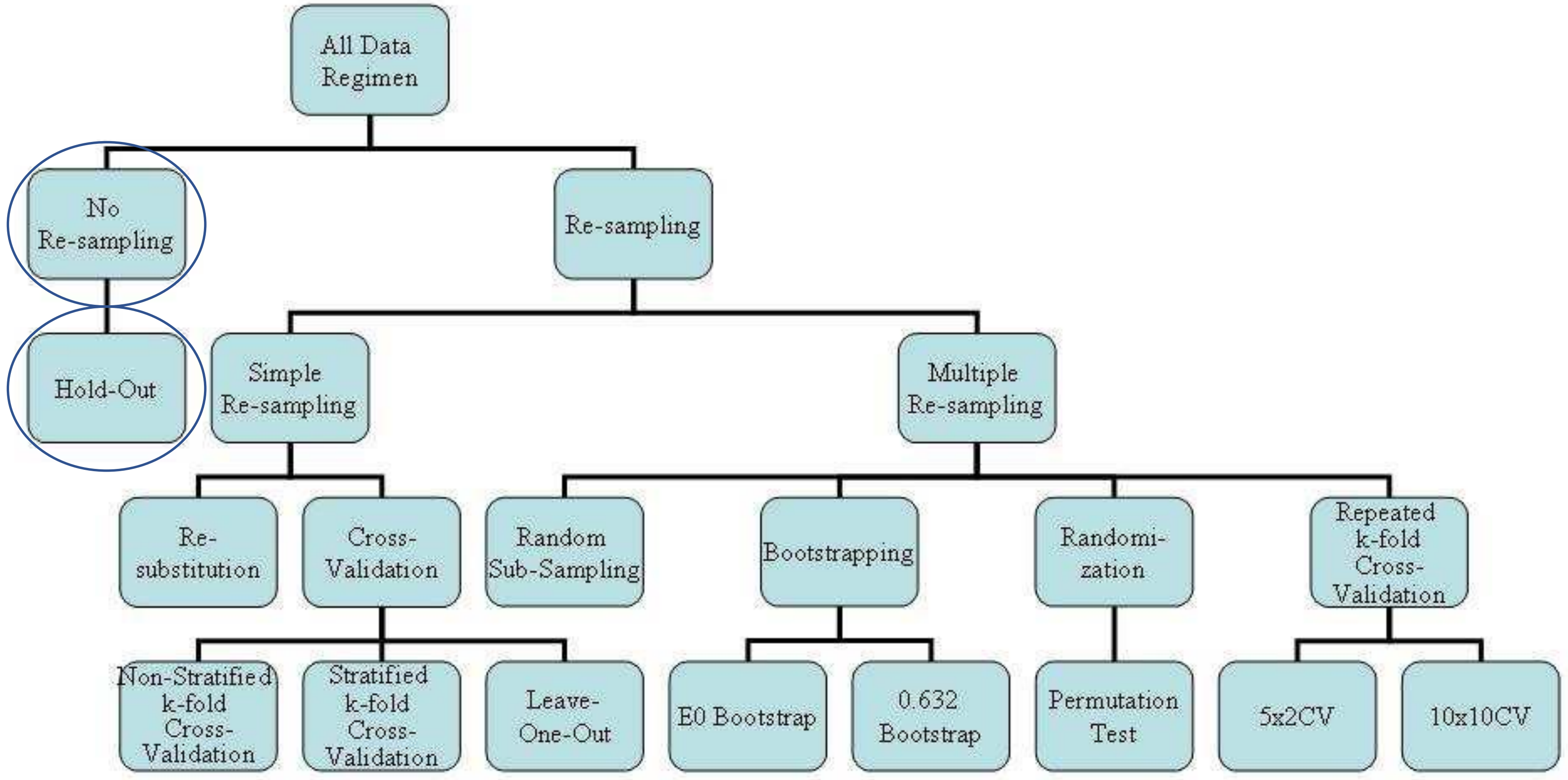
Performance Evaluation

Performance Evaluation

- Dataset Sampling and Re-arrangement
- Performance Metrics
- Performance Charts/Curves

Topic 1:
Dataset Sampling and Arrangement

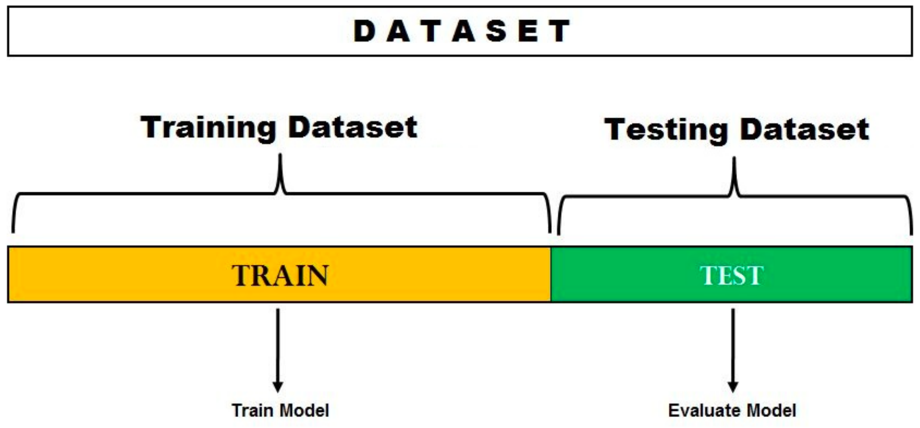
Overview of Re-Sampling Methods

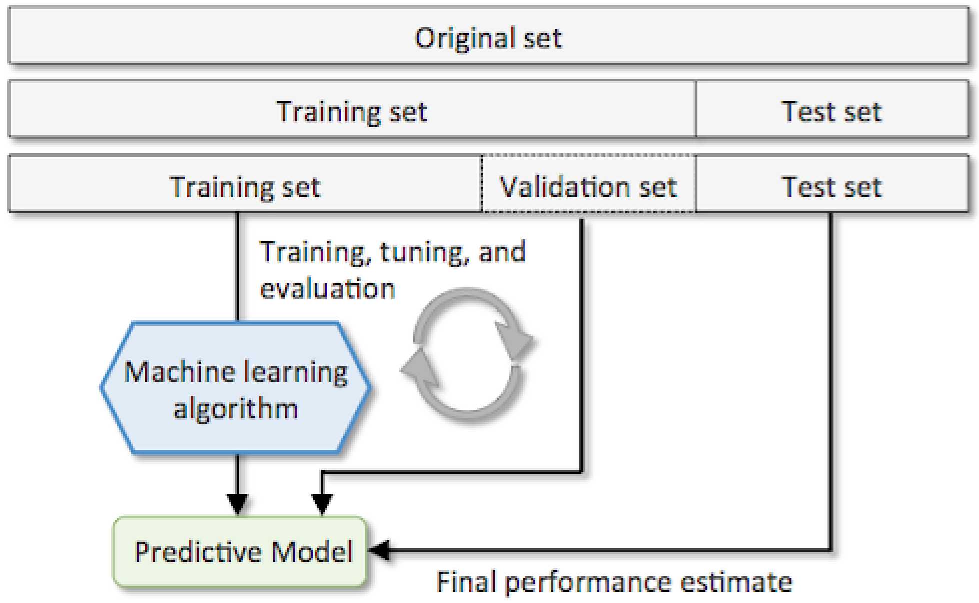


No-Sampling Method: Holdout Methods

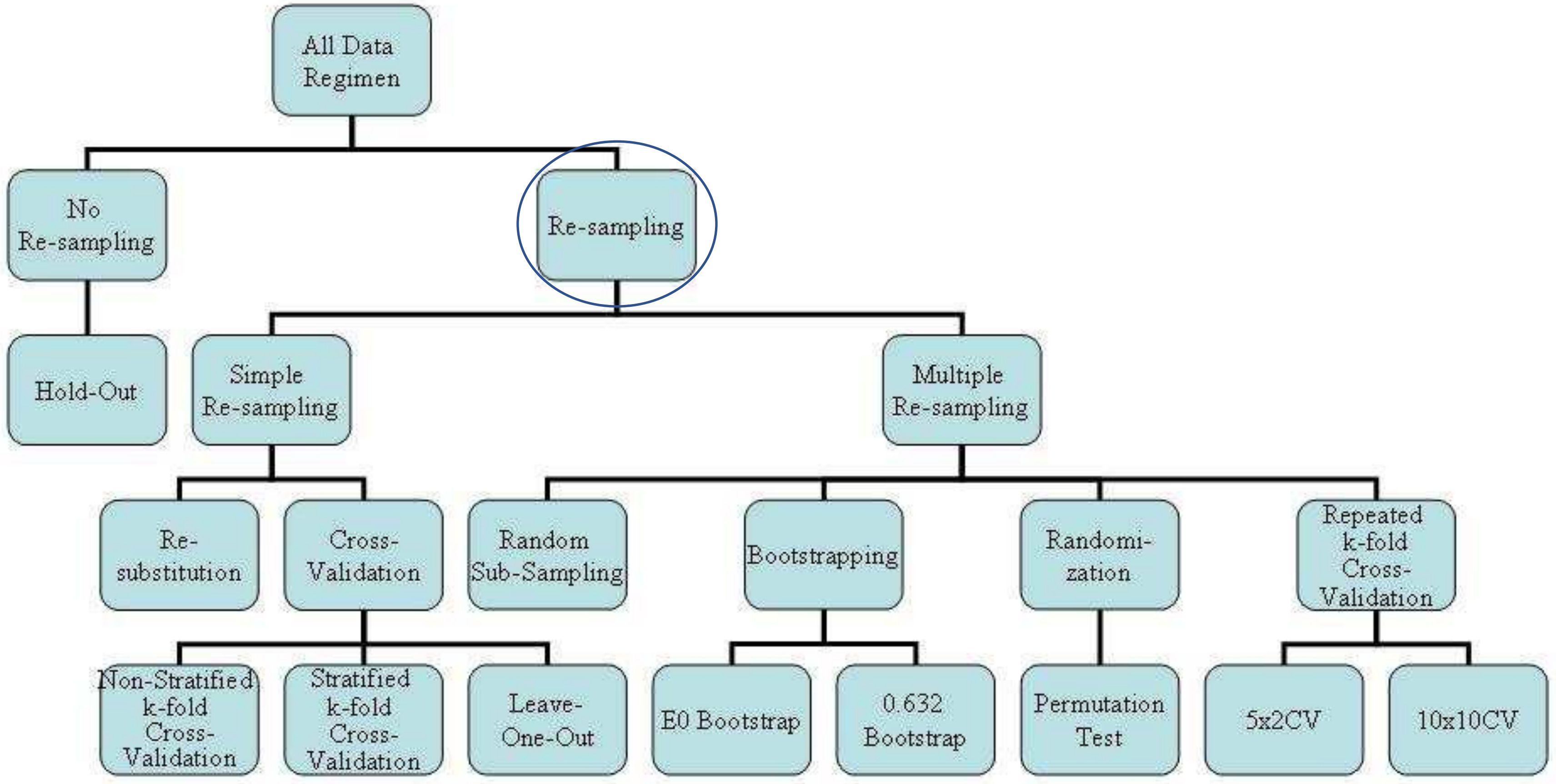
- **No-Sampling:** No explicit or external sampling techniques used. One of the examples is Holdout method.

- **Holdout Method** is the simplest sort of method to evaluate a classifier. In this method, the data set (a collection of data items or examples) is separated into two sets, called the **Training set and Test set**. Typically in 80/20, or 70/30 format.
- When data is sufficiently large, we introduce a validation set, where we typically tune parameters.




Original set
Training set
Test set
Training set
Validation set
Test set
Training, tuning, and evaluation
Machine learning algorithm
Predictive Model
Final performance estimate

Overview of Re-Sampling Methods



What is the Purpose of Resampling?

- Ideally, we would have access to the entire population or a lot of representative data from it.
- This is usually not the case, and the limited data available has to be re-used in clever ways in order to be able to estimate the error of our classifiers as reliably as possible, i.e., to be re-used in clever ways in order to obtain sufficiently large numbers of samples.
- Resampling is divided into two categories:
  - *Simple re-sampling* (where each data point is used for testing only once)
  - *Multiple re-sampling:* (which allows the use of the same data point more than once for testing).

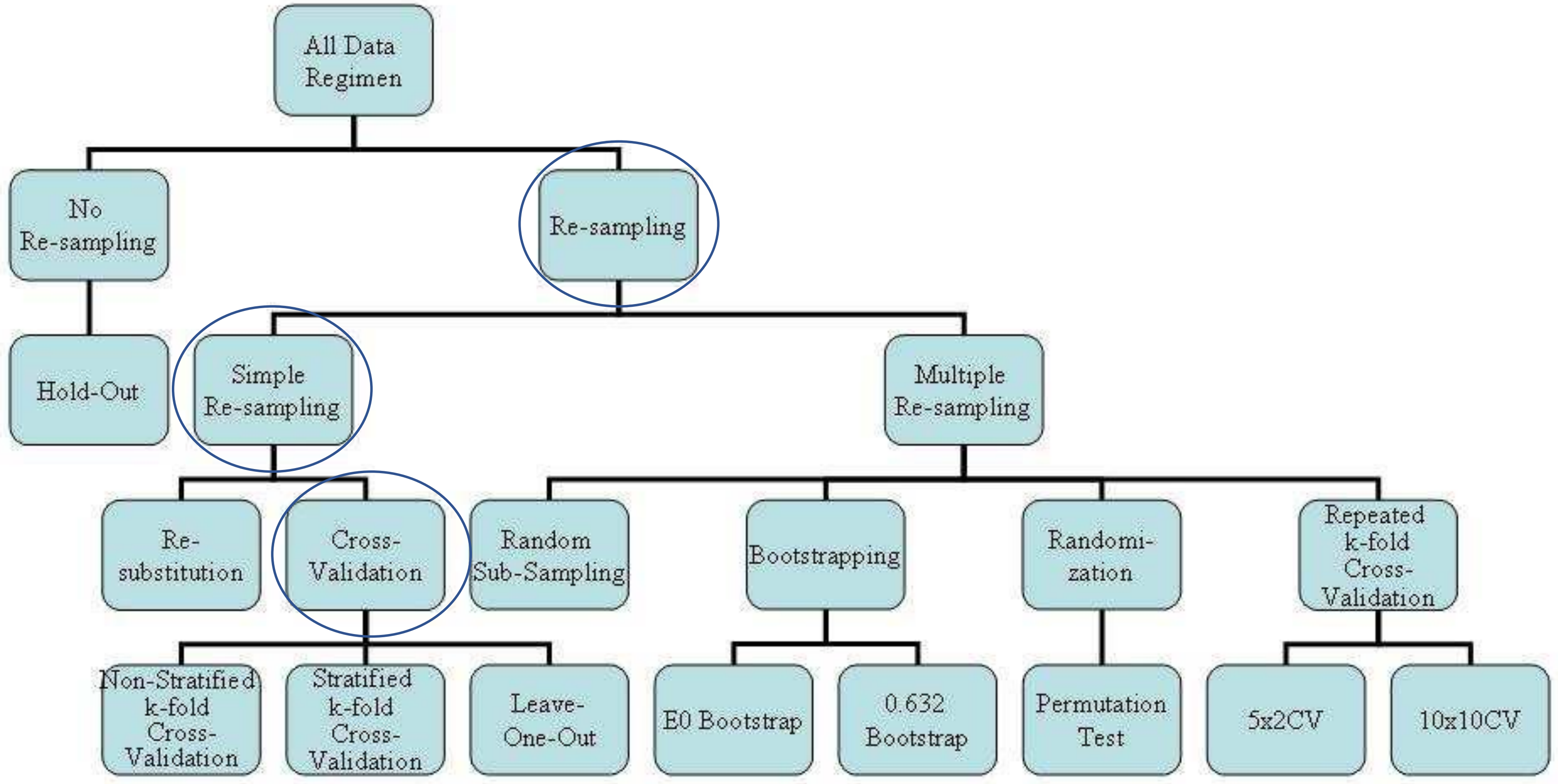
What are the Issues of Resampling?

- Re-sampling is usually followed by Statistical testing. Yet statistical testing relies on the fundamental assumption that the data used to obtain a sample statistics must be independent.
- However, if data is re-used, then this important independence assumption is broken and the result of the statistical test risks being invalid.
- In addition to discussing a few re-sampling approaches, we will underline the issues that may arise when applying them.

Resampling Approaches Discussed in this Tutorial

- Simple Resampling:
  - Cross-Validation and its variants
- Multiple Resampling:
  - The 0.632 Bootstrap
  - The Permutation Test
  - Repeated k-fold Cross-Validation
  - ….

Overview of Re-Sampling Methods

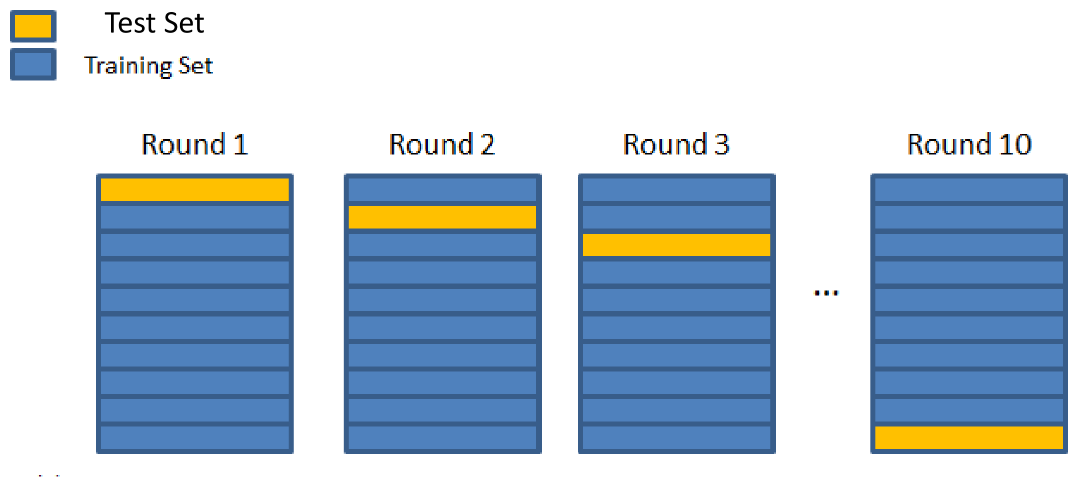


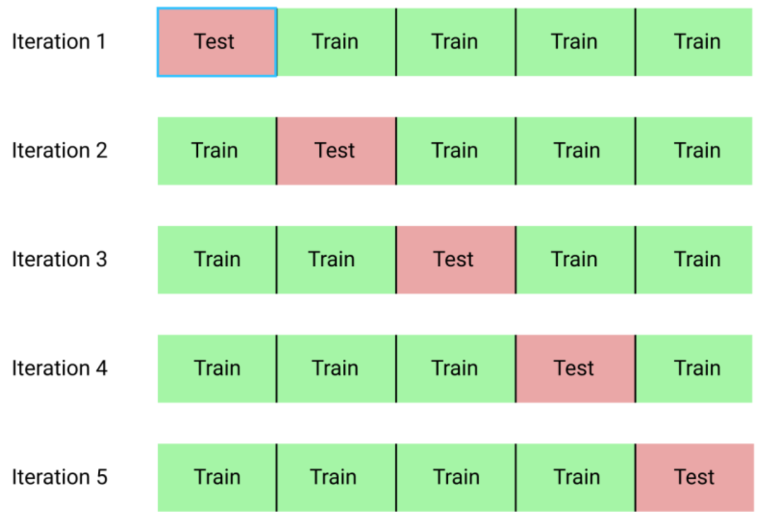
k-fold Cross-Validation

In Cross-Validation, the data set is divided into **k** folds and at each iteration, a different fold is reserved for testing and all the others, used for training the classifiers.

The procedure has a single parameter called **k** that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.
k=10 becoming 10-fold cross-validation.
If k=5 the dataset will be divided into 5 equal parts and the below process will run 5 times, each time with a different holdout set.

Procedure:
1. Take the group as a holdout or test data set
2. Take the remaining groups as a training data set
3. Fit a model on the training set and evaluate it on the test set
4. Retain the evaluation score and discard the model





Issues: k-fold Cross-Validation

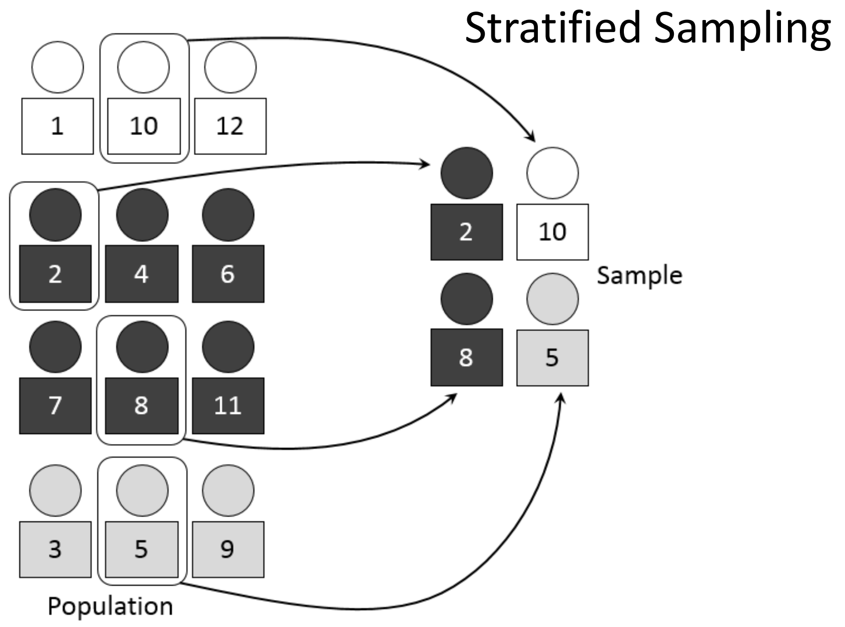
- Using simple K-Fold (non-stratified CV) on a classification problem can be tricky.
- Since we are randomly shuffling the data and then dividing it into folds, chances are we may get highly imbalanced folds which may cause our training to be biased.
- For example, let us somehow get a fold that has majority belonging to one class(say positive) and only a few as negative class.
- This will certainly ruin our training and to avoid this we make stratified folds using stratification.

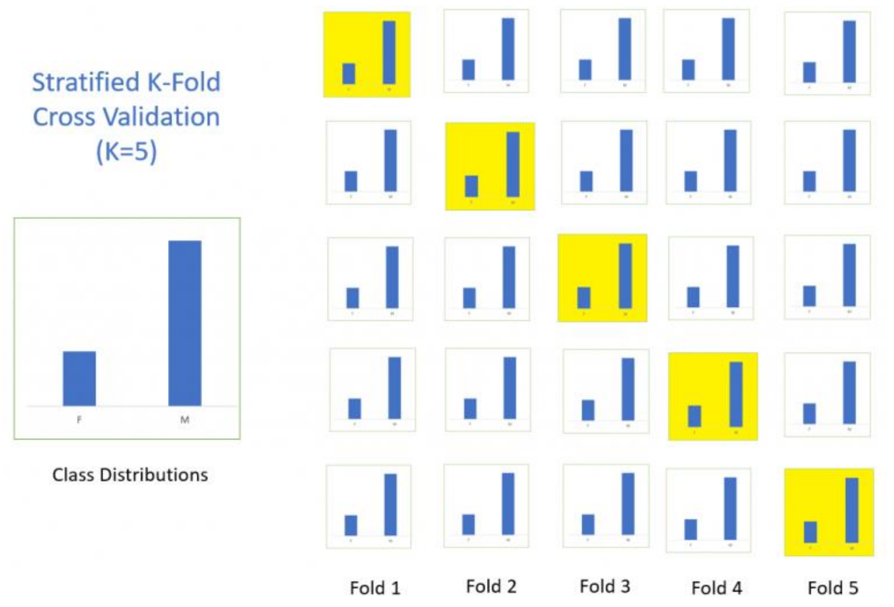
Stratified k-fold Cross-Validation:

Stratified sampling is a sampling technique where the samples are selected in the same proportion (by dividing the population into groups called 'strata' based on a characteristic) as they appear in the population. For example, if the population of interest has 30% male and 70% female subjects, then we divide the population into two ('male' and 'female') groups and choose 30% of the sample from the 'male' group and '70%' of the sample from the 'female' group.

Stratified k-fold Cross-Validation: The splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value. This is called stratified cross-validation.

.

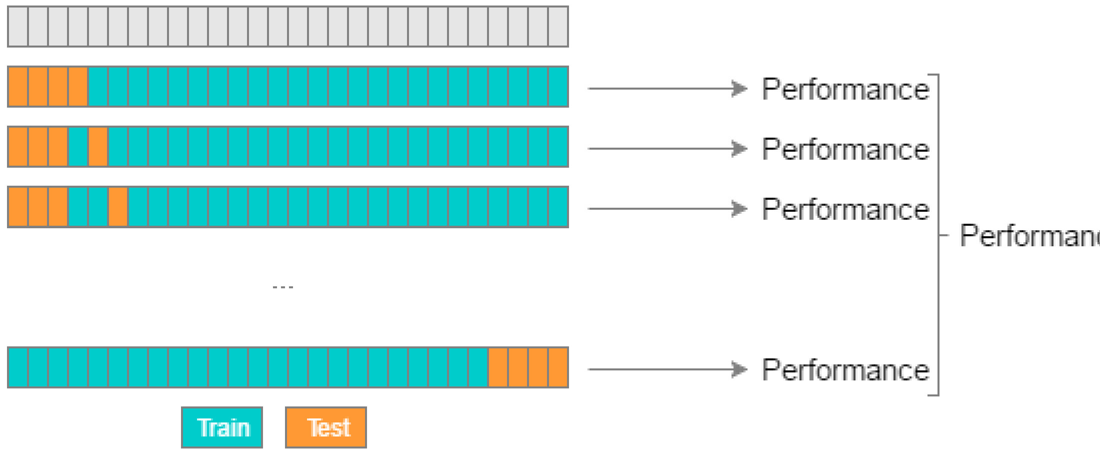




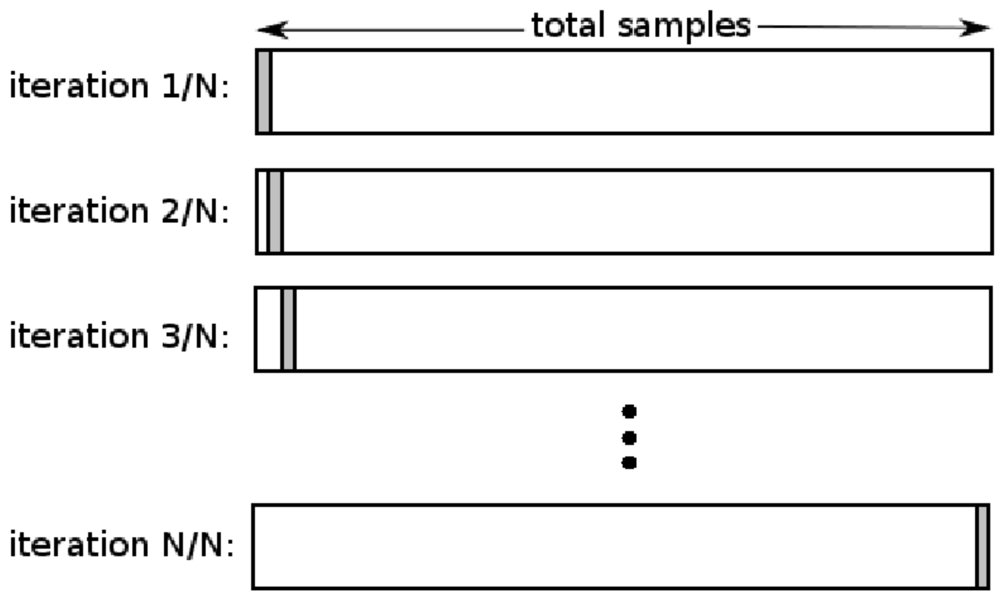
Leave-P-Out cross validation

- When using this exhaustive method, we take **p-number of points out** from the **total number of data (n)**. While training the model we **train it on these (n – p)** data points and **test the model on p data**.
- We repeat this process for **all the possible combinations of *p*** from the original dataset. Then to get the final accuracy, we average the accuracies from all these iterations.
- This is an exhaustive method as we train the model on every possible combination of data points. Remember if we choose a higher value for p, then the number of combinations will be more and we can say the method gets a lot more exhaustive.
- **Leave-one-out cross validation**

  This is a simple variation of Leave-P-Out cross validation and the value of p is set as 1. This makes the method much less exhaustive as now for n data points and p = 1, we have n number of combinations.



Leave-P-out cross validation



Leave-one-out cross validation

Considerations about k-fold Cross-Validation and its variants

- k-fold Cross-Validation is the best known and most commonly used resampling technique.
- K-fold Cross-Validation is less computer intensive than Leave-One-Out.
- The testing sets are independent of one another, as required, by many statistical testing methods, but the training sets are highly overlapping. This can affect the bias of the error estimates.
- Leave-One-Out produces error estimates with high variance given that the testing set at each fold contains only one example. The classifier is practically unbiased since each fold trains on almost all the data.

Topic 2:
Performance Metrics

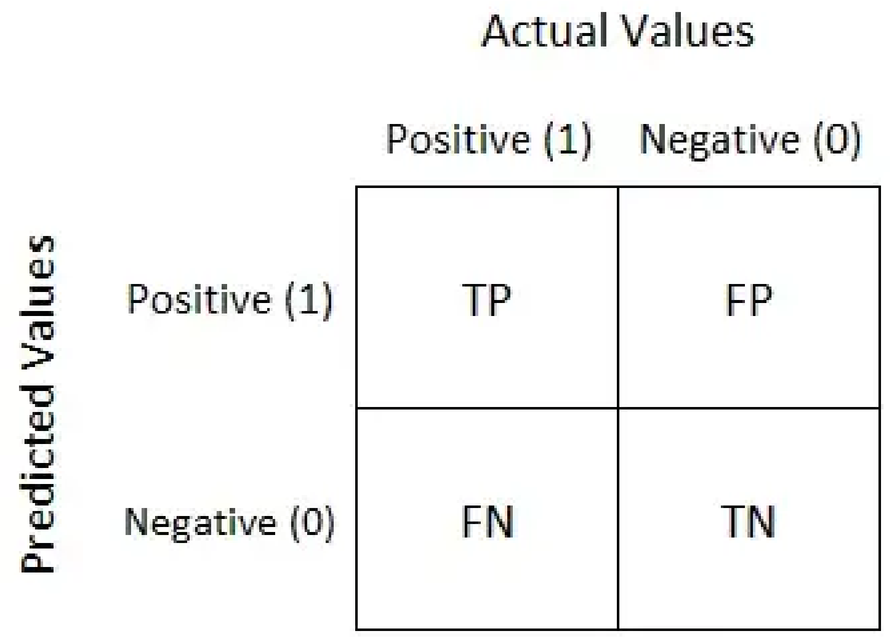
Metrics

It is extremely important to use **quantitative metrics** for evaluating a machine learning model

- Until now, we relied on the **cost function value** for regression and classification
- Other metrics can be used to **better evaluate** and understand the model
- **For classification**
  - ✓ Accuracy/Precision/Recall/F1-score, ROC curves,…
- **For regression**
  - ✓ Normalized RMSE, Normalized Mean Absolute Error (NMAE),…

Confusion matrix

- A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

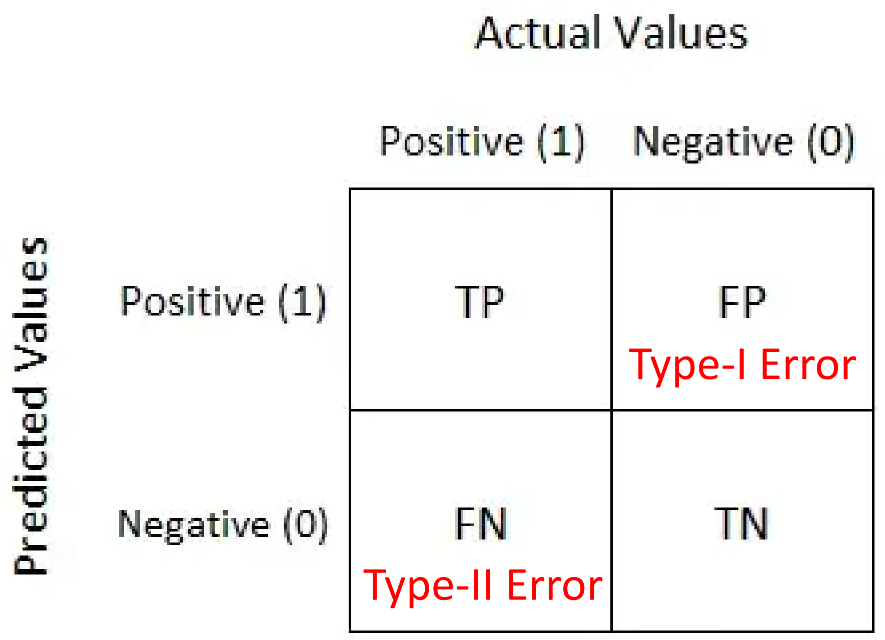
TP, TN, FP, FN

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.
False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP Type-I Error
	Negative (0)	FN Type-II Error	TN

Find out the values of TP, TN, FP, FN

1=Positive class
0=Negative Class

Data	Actual	Predicted
1	1	1
2	1	1
3	0	1
4	0	1
5	1	0
6	0	0
7	1	1

TP =

TN =

FP =

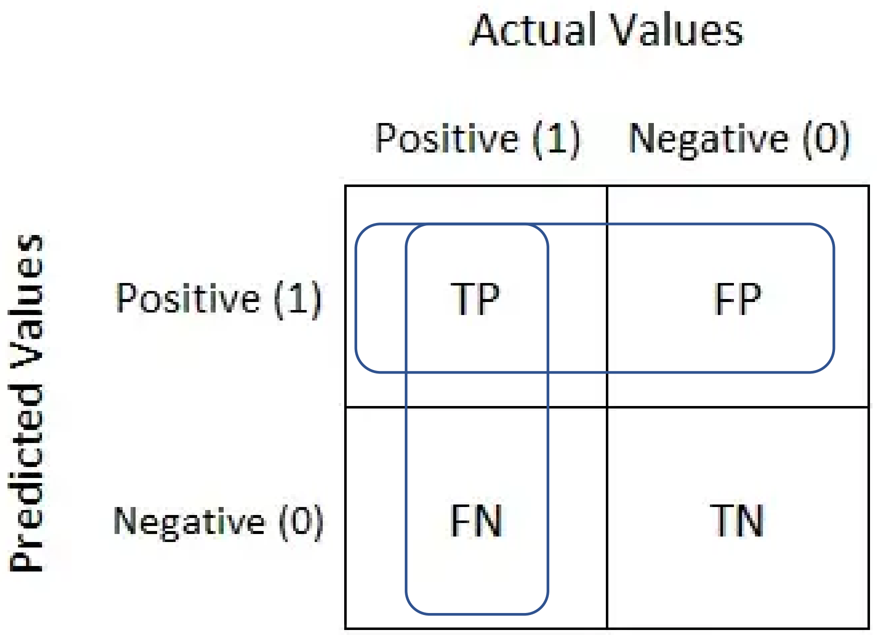
FN =

Accuracy

- Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. For a model, if we have got 0.803 means our model is approx. 80% accurate.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

- Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.



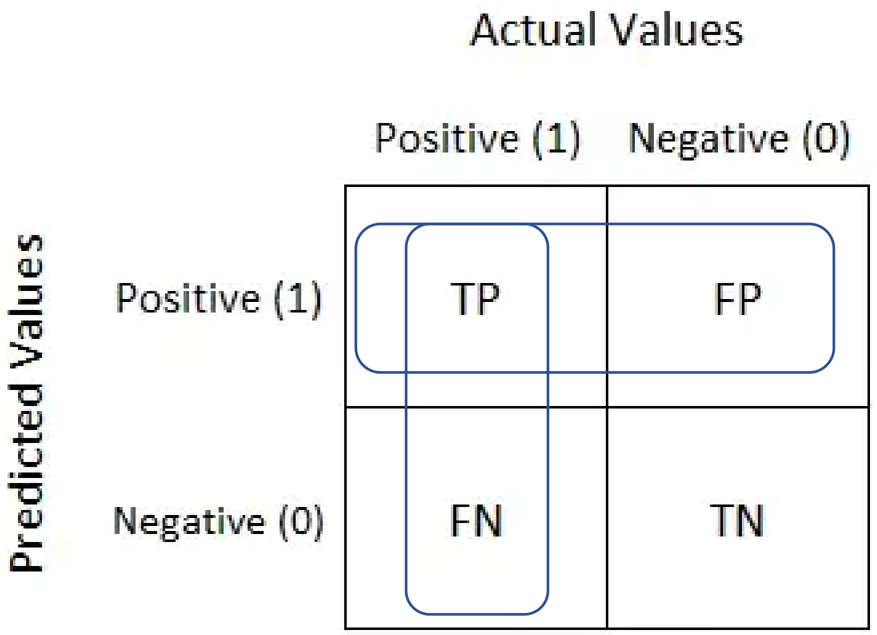
Precision, Recall

- **Precision** - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. For ex, a 0.788 precision which is pretty good.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? For ex, if we get recall of 0.631 which is good for this model as it's above 0.5.

$$Recall = \frac{TP}{TP + FN}$$



F1-score

- F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. A F1 score is 0.701 is a good indicator of performance.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F_β Score

- $F_\beta = (1+\beta^2)\frac{Precission \times Recall}{\beta^2(Precission + Recall)}$

If β=1, it is $F1 - Score$
β=0.5, it is $F0.5 - Score$
β=2, it is $F2 - Score$

When β = 1 $F_1 = (1+1^2)\frac{Precission \times Recall}{1^2 * (Precission + Recall)}$

$$F_1 = 2\frac{Precission \times Recall}{Precission + Recall}$$

It’s a harmonic mean of precision and recall

Use- where FP and FN both are important.

When FP (Type-I Error) is having more impact than FN, its better to reduce the value of $\boldsymbol{\beta}$.
Typically $\boldsymbol{\beta}$=0.5.

When FN (Type-II Error) is having more impact than FP, its better to increase the value of $\boldsymbol{\beta}$.
Typically $\boldsymbol{\beta}$=2.

Practice Math

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP 30	FP 5
	Negative (0)	FN 10	TN 55

Accuracy = (55 + 30)/(55 + 5 + 30 + 10) = 0.85

precision = 30/(30+ 5) = 0.857

Recall = 30/(30+ 10) = 0.75

F1 Score = 2* (0.857 * 0.75)/(0.857 + 0.75) = 0.799.

At a glance

Recall Sensitivity True positive rate (TPR)	$\frac{TP}{FN+TP} = \frac{TP}{P}$
False positive rate (FPR) False alarm rate	$\frac{FP}{TN+FP} = \frac{FP}{N}$
Specificity True negative rate (TNR)	$\frac{TN}{TN+FP} = \frac{TN}{N} = 1 - FPR$
Precision	$\frac{TP}{TP+FP}$
False negative rate (FNR)	$\frac{FN}{FN+TP} = \frac{FN}{P}$
Accuracy	$\frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Root-Mean Squared Error (RMSE)

- The Root-Mean Squared Error (RMSE) is usually used for **regression**, but can also be used with probabilistic classifiers. The formula for the RMSE is:

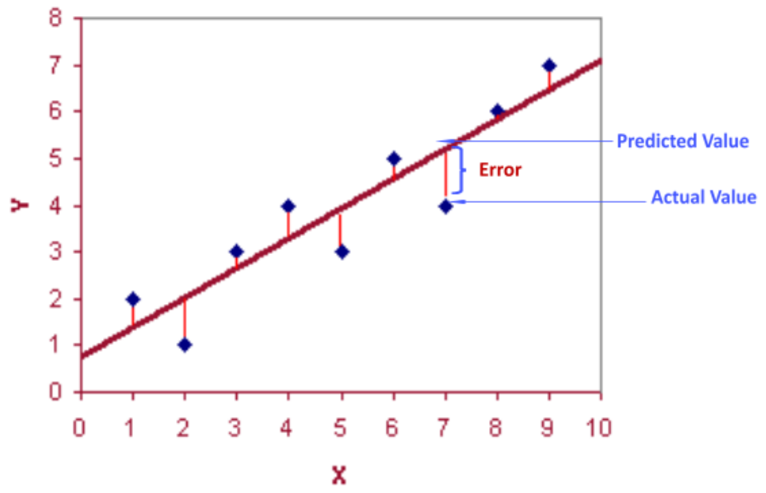
$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Where,
n =total number of test examples
$\hat{y}$ = predicted output
y_i = the actual label.

No	$\hat{y}$	y_i	$(\hat{y} - y_i)^2$
1	.95	1	.0025
2	.6	0	.36
3	.8	1	.04
4	.75	0	.5625
5	.9	1	.01

$$\text{RMSE(f)} = \frac{\sqrt{(.0025+.36+.04+.5625+.01)}}{5} = 0.4416$$

RMSE calculates total error in a prediction


Predicted Value
Error
Actual Value
Y
X