

## Evaluation

### In a nutshell



The diagram illustrates the machine learning workflow. A 'Data generating process'  $P_{xy}$  provides samples to both a training set  $D_{train}$  and a test set  $D_{test}$ . The training set  $D_{train}$  is used by a 'Learner' to 'Fit' a 'Model  $\hat{f}(x)$ '. The 'Model  $\hat{f}(x)$ ' is then used to 'Predict' on the test set  $D_{test}$ , which contains labels  $y$  and  $\hat{y}$ . The difference between  $y$  and  $\hat{y}$  is labeled 'Loss?'.

- Understand what the Generalization Error is
- Get an overview on how we evaluate performance of learners
- Learn about some evaluation metrics
- Understand why we do resampling

# ESTIMATING THE GENERALIZATION ERROR

- For a fixed model, we are interested in the Generalization Error (GE):  $\text{GE}(\hat{f}, L) := \mathbb{E} \left[ L(y, \hat{f}(\mathbf{x})) \right]$ , i.e. the expected error the model makes for data  $(\mathbf{x}, y) \sim \mathbb{P}_{xy}$ .
- We need an estimator for the GE with  $m$  test observations:

$$\widehat{\text{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x}, y)} \left[ L(y, \hat{f}(\mathbf{x})) \right]$$

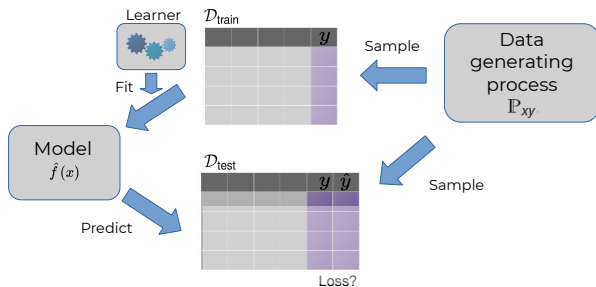
- However, if  $(\mathbf{x}, y) \in \mathcal{D}_{\text{train}}$ ,  $\widehat{\text{GE}}(\hat{f}, L)$  will be biased via overfitting the training data.
- Thus, we estimate the GE using unseen data  $(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}$ :

$$\widehat{\text{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}} \left[ L(y, \hat{f}(\mathbf{x})) \right]$$



# ESTIMATING THE GENERALIZATION ERROR

- Usually, we have no access to new **unseen** data.
- Thus, we divide our data set manually into  $\mathcal{D}_{\text{train}}$  and  $\mathcal{D}_{\text{test}}$ .
- This process is depicted below.





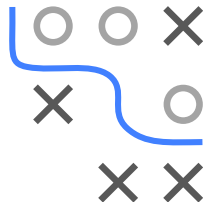
# DEEP DIVE: PROPERNESS

- A scoring rule  $\mathbf{S}$  is proper relative to  $\mathcal{F}$  if (where a low value of the scoring rule is better):

$$\mathbf{S}(Q, Q) \leq \mathbf{S}(F, Q) \forall F, Q \in \mathcal{F}$$

with  $\mathcal{F}$  being a convex class of probability measures.

- This means that a scoring rule should be optimal for the actual data target distribution, i.e. we are rewarded for properly modeling the target.



## METRICS FOR CLASSIFICATION

Commonly used evaluation metrics include:

- Accuracy:

$$\rho_{ACC} = \frac{1}{m} \sum_{i=1}^m [y^{(i)} = \hat{y}^{(i)}] \in [0, 1].$$

- Misclassification error (MCE):

$$\rho_{MCE} = \frac{1}{m} \sum_{i=1}^m [y^{(i)} \neq \hat{y}^{(i)}] \in [0, 1].$$

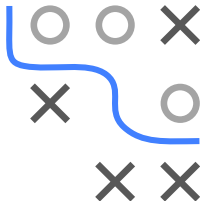
- Brier Score:

$$\rho_{BS} = \frac{1}{m} \sum_{i=1}^m (\hat{\pi}^{(i)} - y^{(i)})^2$$

- Log-loss:

$$\rho_{LL} = \frac{1}{m} \sum_{i=1}^m \left( -y^{(i)} \log(\hat{\pi}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{\pi}^{(i)}) \right).$$

The probabilistic metrics, Brier Score and Log-Loss penalize false confidence, i.e. predicting the wrong label with high probability, heavily.



# METRICS FOR CLASSIFICATION

For hard-label classification, the confusion matrix is a useful representation:

		True Class $y$	
		+	-
Pred.	+	True Positive (TP)	False Positive (FP)
$\hat{y}$	-	False Negative (FN)	True Negative (TN)



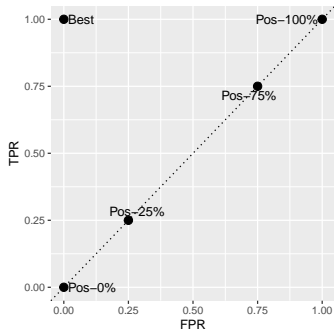
From this matrix a variety of evaluation metrics, including precision and recall, can be computed.

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

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

# RECEIVER OPERATING CHARACTERISTICS

- Receiver operating characteristics (ROC) performs evaluation for binary classifiers beyond single metrics.
- We can assess classifiers by their TPR (y-axis) and FPR (x-axis).
- We aim to identify good classifiers who (weakly) dominate others.
- For example, the "Best" classifier in the image strictly dominates "Pos-25%" and "Pos-75%" and weakly dominates the others.

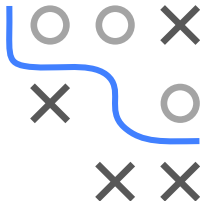




# METRICS FOR REGRESSION

Commonly used evaluation metrics include:

- Sum of Squared Errors (SSE):  $\rho_{SSE}(\mathbf{y}, \mathbf{F}) = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2$
- Mean Squared Error (MSE):  $\rho_{MSE}(\mathbf{y}, \mathbf{F}) = \frac{1}{m} \sum_{i=1}^m SSE$
- Root Mean Squared Error (RMSE):  $\rho_{RMSE}(\mathbf{y}, \mathbf{F}) = \sqrt{MSE}$
- R-Squared:  $\rho_{R^2}(\mathbf{y}, \mathbf{F}) = 1 - \frac{\sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^m (y^{(i)} - \bar{y})^2}$
- Mean Absolute Error (MAE):  
 $\rho_{MAE}(\mathbf{y}, \mathbf{F}) = \frac{1}{m} \sum_{i=1}^m |y^{(i)} - \hat{y}^{(i)}| \in [0; \infty)$



## ESTIMATING THE GENERALIZATION ERROR (BETTER)

## While

$$\widehat{\text{GE}}(\hat{f}, L) := \frac{1}{m} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}} \left[ L(y, \hat{f}(\mathbf{x})) \right]$$

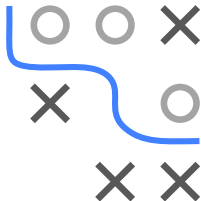
will be unbiased, with a small  $m$  it will suffer from high variance. We have two options to decrease the variance:

- Increase  $m$ .
- Compute  $\widehat{\text{GE}}(\hat{f}, L)$  for multiple test sets and aggregate them.

With a finite amount of data, increasing  $m$  would mean to decrease the size of the training data. Thus, we focus on using multiple ( $B$ ) test sets:

$$\mathcal{J} = ((J_{\text{train},1}, J_{\text{test},1}), \dots, (J_{\text{train},B}, J_{\text{test},B})).$$

where we compute  $\widehat{\text{GE}}(\hat{f}, L)$  for each set and aggregate the estimates. These  $B$  sets are generated through **resampling**.



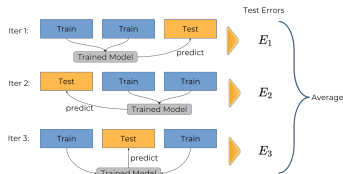
## RESAMPLING

There exist a few well established resampling strategies:

- (Repeated) Hold-out / Subsampling
- Cross validation
- Bootstrap

All methods aim to generate  $\mathcal{J}$  by splitting the full data set (repeatedly) into a train and test set. The model is trained on the respective train set and evaluated on the test set.

### Example: 3-fold cross validation



In order to robustify performance estimates we can repeat these resamplings, e.g. we could perform 10 times 8 fold cross validation.

