REDUCE

Evaluating Model Performance

H.Nina del Rio-Ares

Institute of Marine Research (IIM-CSIC)

CSIC

INSTITUTO DE INVESTIGACIÓNS MARIÑAS

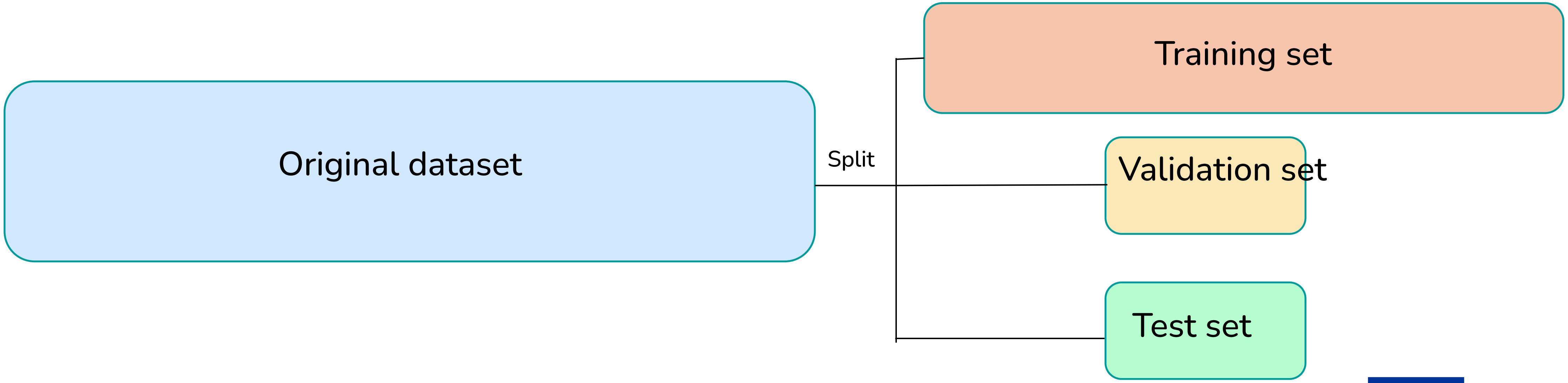
Horizon Europe GA no. 101135583

Funded by
the European Union

Dataset split

1. Training, Validation and Test

- **Training set**: Is the subset of the original dataset that we use to train our model
- **Validation set**: Subset used to evaluate the model performance during training
- **Test set**: Subset used for the final assessment of the model's performance



Funded by the European Union

2. Cross-Validation (CV)

- **Problem**: Partitioning our dataset into three subsets drastically **reduce the number of samples** which can be used for learning the model, and the results can depend on a particular random choice of the training and validation sets.

Particularly if our dataset is small!

How can we solve this problem?

Funded by the European Union

2. Cross-Validation (CV)

✖ **Problem**: Partitioning our dataset into three subsets drastically **reduce the number of samples** which can be used for learning the model, and the results can depend on a particular random choice of the training and validation sets.

Particularly if our dataset is small!

How can we solve this problem?

CROSS-VALIDATION
Is a technique which consists of splitting the dataset into several subsets; the model is trained on some and validated on the others, repeating the process multiple times.

Funded by the European Union

Cross-Validation

3. k-fold

How it works? 🤔

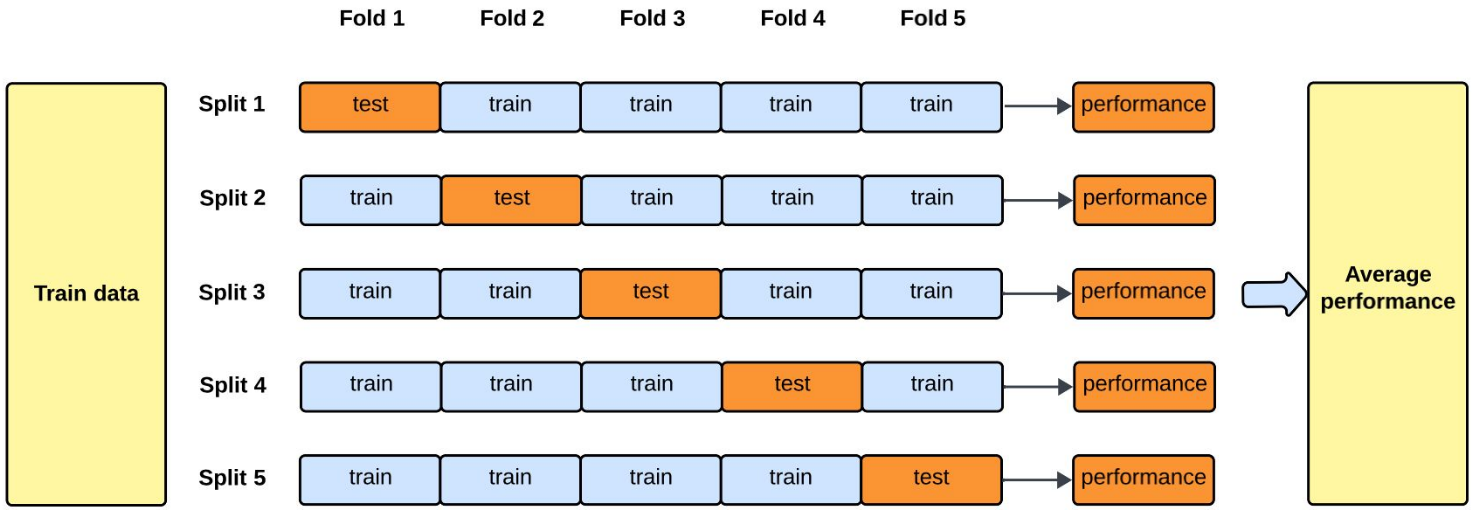
1. Split the dataset into **k equal parts** (called folds).

2. For each iteration:
- Train on k-1 folds
- Validate on the **remaining** fold

3. Repeat this **k times**, using a different fold for validation each time

4. Average the results.

5. Train on **k – 1** folds.



Standard k-fold cross validation.

Image Source: Image by author, en Medium

Funded by the European Union

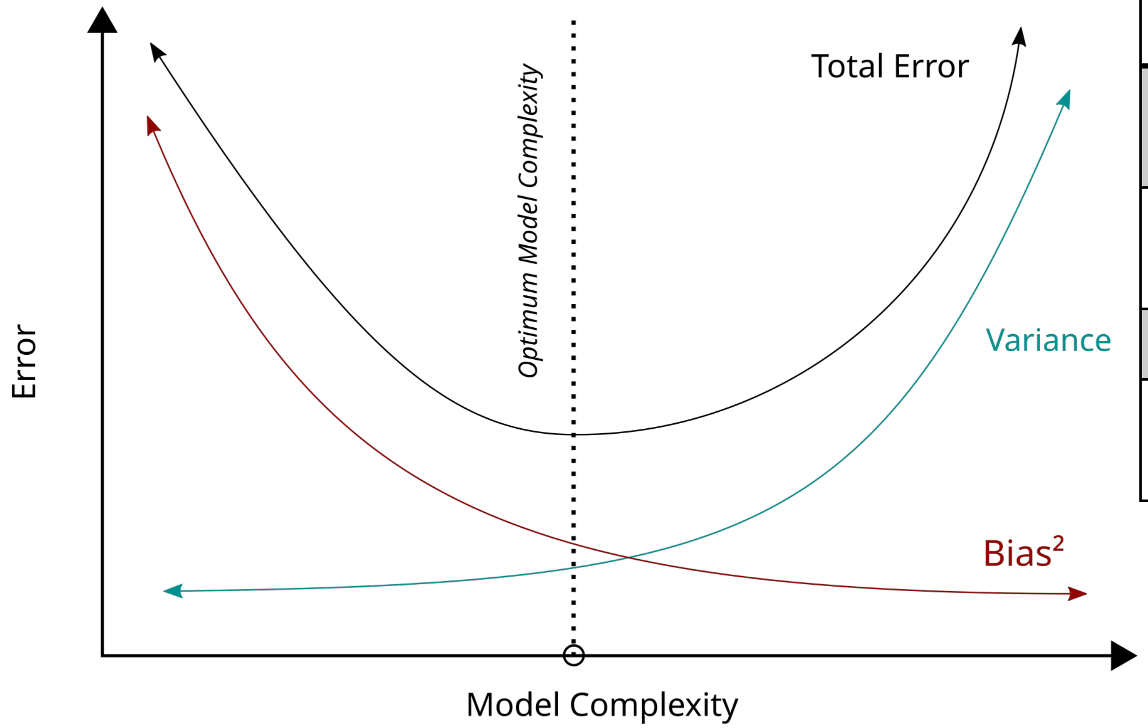
4. CV Pros and Cons

✅ **Advantages**	⚠️ **Disadvantages / Challenges**
Reduces the risk of biased results from a single train/val split	More computationally expensive
All data are used for both training and validation	Potential **overfitting** on small datasets
More robust model selection	Can be complex to implement
Maximizes the amount of training data available	

Funded by the European Union

Overfitting

5. Bias-Variance tradeoff



Concept	High Bias	High Variance
What happens?	Model is too simple	Model is too complex/flexible
Issue	Misses important patterns	Captures noise and irrelevant details
Consequence	**Underfitting**	**Overfitting**
Error	High on both training and test data	Low on training, high on test data

"Bias and variance contributing to total error." *Wikipedia*, https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff#/media/File:Bias_and_variance_contributing_to_total_error.svg, accessed April 14, 2025.

Funded by the European Union

6. Techniques to Prevent Overfitting

Overfitting ⟶ when the model performs well on the on the training but is not capable of generalizes on knew data.

1. **Cross Validation** 😎 Now we know what this is!

2. **Data Augmentation**

Data augmentation is a technique which consist on **increasing the training set** by creating knew images from the ones that we already have.

3. **Dropout**

Dropout is a regularization technique used during training in neural networks to reduce overfitting. It works by randomly "turning off" a percentage of neurons in a given layer during each training iteration.

Let's learn more about this! ⟶

Funded by the European Union

Techniques to Prevent Overfitting

7. Data Augmentation

It's very useful if we have a small training data set. This data transformation can be:

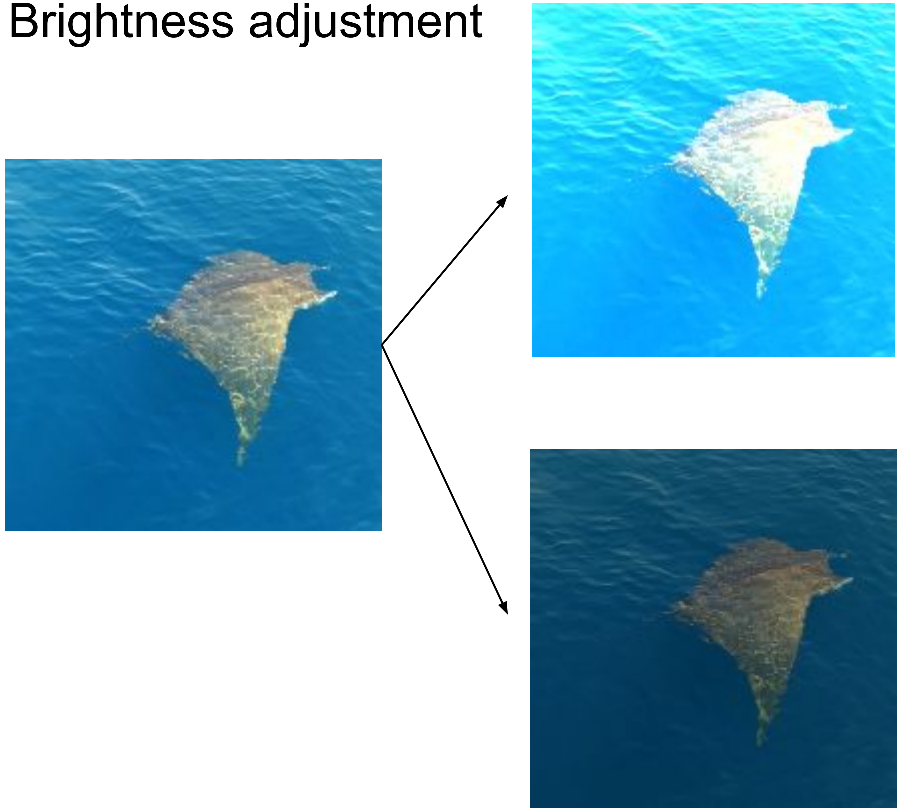
- **Geometric Transformations**: translation, rotation, flipping (horizontal and vertical), cropping…
- **Color Space Transformations**, modify the color properties of images, addressing variations in lighting (brightness adjustment, contrast adjustment, saturation, color Jittering)
- **Noise Injection:** Gaussian Noise, DropBlcock (adding gaussian noise), salt and pepper noise…
- **Mosaic Augmentation (YOLOv5, YOLOv8):** Combine 4 different images into a single mosaic.

Funded by the European Union

Techniques to Prevent Overfitting

7. Data Augmentation

Brightness adjustment



Horizontal and vertical flipping

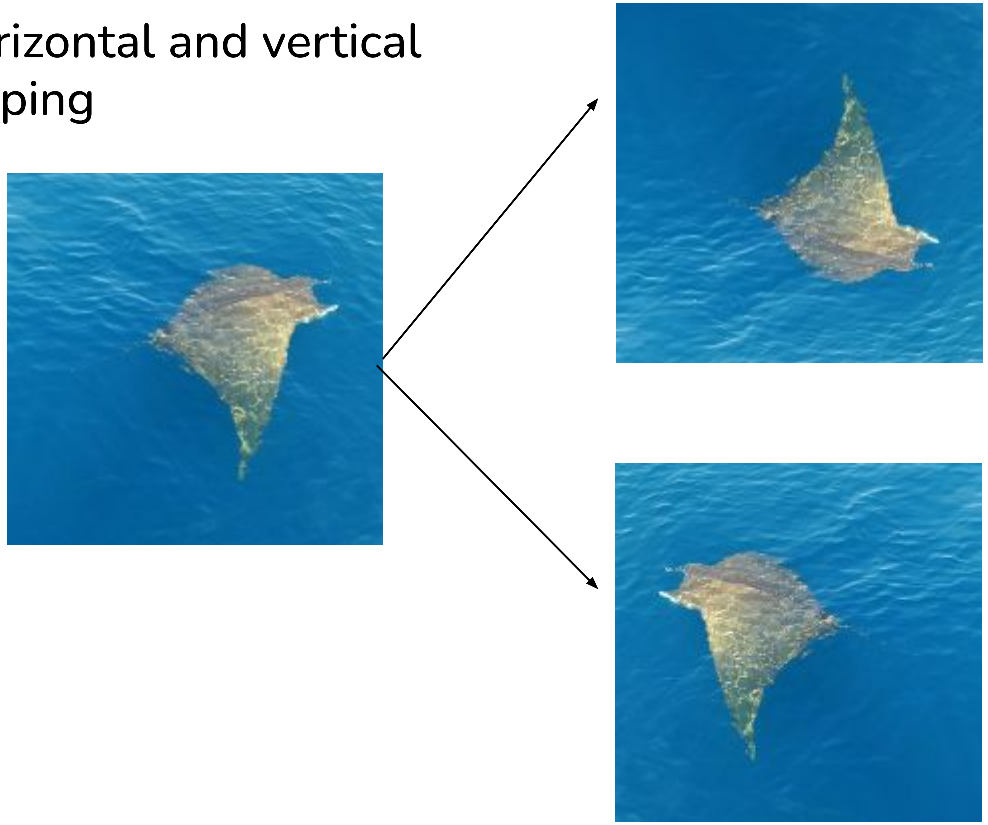


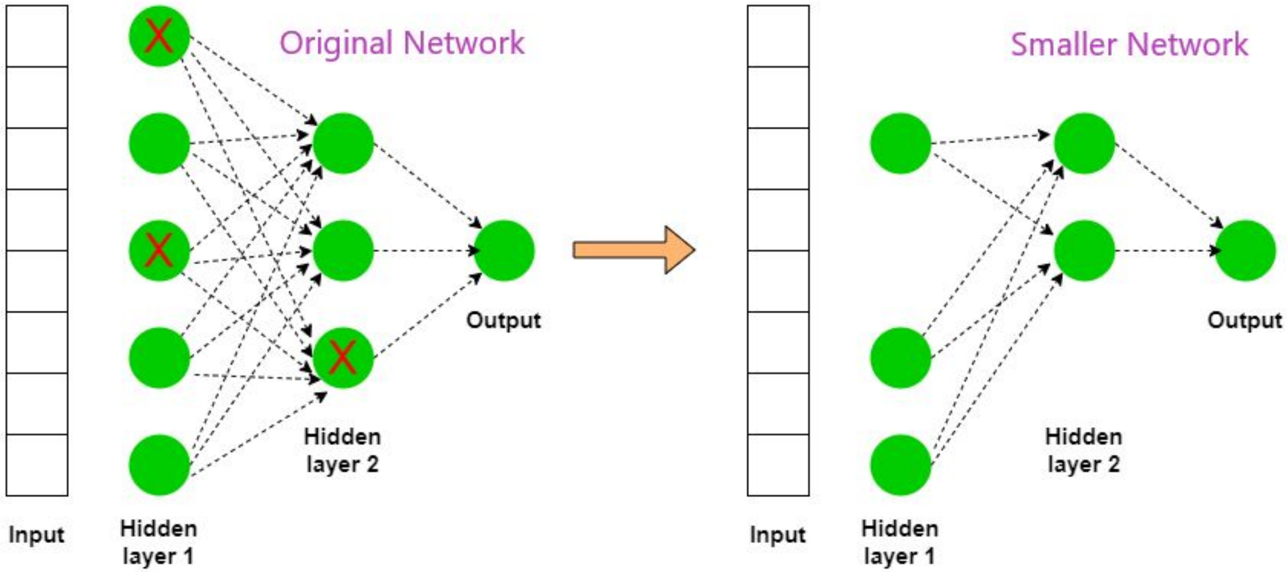
Image source: Mobula tarapacana (Philippi, 1892), observed in the United States by *sweiss243*, published on iNaturalist, licensed under CC BY-NC 4.0. Original photo: https://www.inaturalist.org/photos/292043233

Funded by the European Union

Techniques to Prevent Overfitting

8. Dropout

During the training, randomly a percentage of neurons is deactivated in a layer during each forward pass. This means that these neurons are temporarily ignored, and do not participate in forward or backward passes. By randomly deactivating neurons, the network is prevented from becoming too reliant on any single feature or path through the model. **This encourages the learning of more robust and generalizable patterns.**



Dropout regularization process

Image Source: Image by author, made with draw.io, en Medium

Funded by the European Union

Techniques to Prevent Overfitting

8. Dropout

Dropout: A simple Way to Prevent Neural Networks from Overfitting

Journal of Machine Learning Research 15 (2014) 1929-1958 Submitted 11/13; Published 6/14

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

Editor: Yoshua Bengio

Abstract

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time,

Funded by the European Union

Performance Metrics

9. Confusion Matrix

Actual\Predicted	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Example:

Actual\Predicted	Shark	Turtle	Ray
Shark	50	5	10
Turtle	3	60	2
Ray	5	4	45

1. 50 sharks were correctly classified as sharks, 60 turtles were correctly classified as turtles and 45 rays were correctly classified as rays.
2. The values off the diagonal indicate classification errors. For example, 5 sharks were misclassified as turtle and 10 as rays

Funded by the European Union

Performance Metrics

10. ACCURACY

$$\text{Accuracy} = \underbrace{(TP + TN)}_{\text{Total correct predictions}}/\overbrace{(TP + TN + FP + FN)}^{\text{All predictions}}$$

Actual\Predicted	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Funded by
the European Union

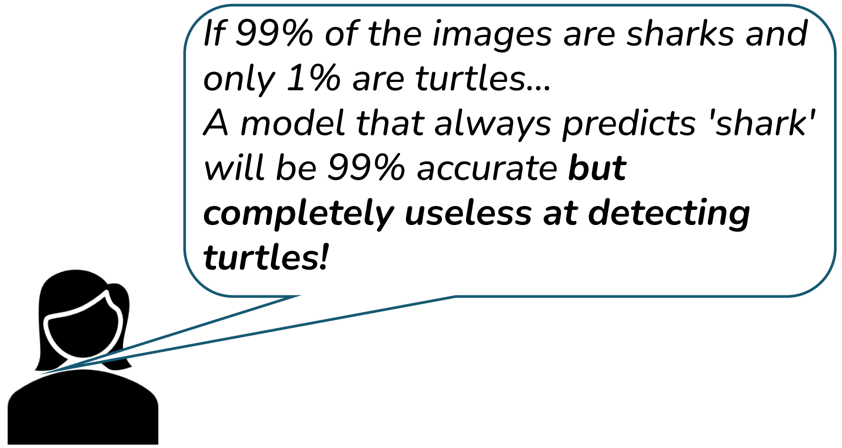
Performance Metrics

10. ACCURACY

$$\text{Accuracy} = \underbrace{(TP + TN)}_{\text{Total correct predictions}} / \overbrace{(TP + TN + FP + FN)}^{\text{All predictions}}$$

Actual\Predicted	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Accuracy can be misleading when working with imbalanced dataset!



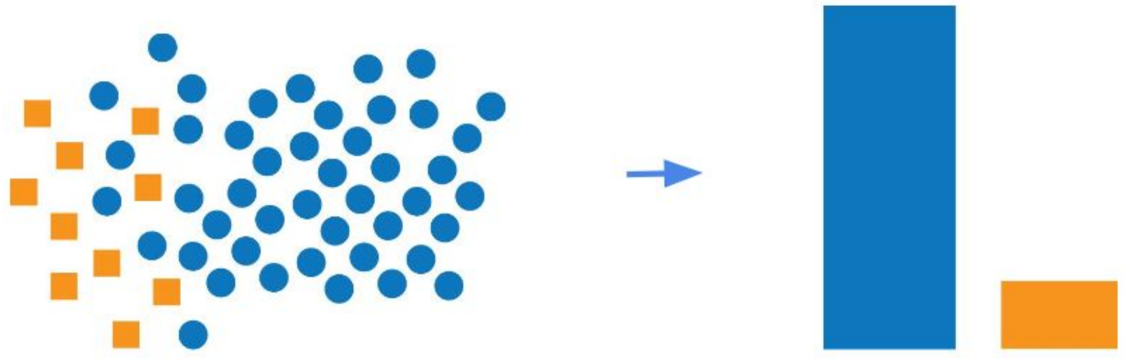


Image source: **M. Farhan Tandia**, *"Some tricks in handling imbalanced dataset"*, LinkedIn, link

Funded by the European Union

Performance Metrics

11. PRECISION AND RECALL

$$\text{Precision} = TP \,/\, \underbrace{(TP + FP)}_{\text{Positive predictions}}$$

$$\text{Recall} = TP \,/\, \underbrace{(TP + FN)}_{\text{Actual positives}}$$

Actual\Predicted	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Funded by the European Union

11. PRECISION AND RECALL

Precision = $TP \,/\, \underbrace{(TP + FP)}_{\text{Positive predictions}}$

Recall = $TP \,/\, \underbrace{(TP + FN)}_{\text{Actual positives}}$

EXAMPLE: Trade-off in shark classification.

Model	Precision	Recall	Interpretation
A	0.95	0.4	Very accurate when predicting shark, but misses many real sharks (FN)
B	0.30	0.99	Detects nearly all sharks, but often mistakes other species for sharks (FP)
C	0.80	0.82	Balanced performance

Choosing between precision and recall depends on your goal, but in most cases, we need a **balance between both metrics**.

HOW? 🤔

A metric which combines the two values ⇒ F-score

Funded by the European Union

Performance Metrics

12. F-SCORE

$$F_\beta = (1 + \beta^2)\frac{precision \times recall}{\beta^2(precision + recall)}$$

Controls the importance given to precision or recall

Special case: $\beta = 1$

The harmonic mean of precision and recall balancing both metrics

$$F_1 = 2\frac{precision \times recall}{(precision + recall)}$$

WHY the harmonic mean? Because it punishes unbalanced values

Funded by the European Union

12. F-SCORE

$$F_1 = 2\frac{precision \times recall}{(precision + recall)}$$

How it works in a **multiclass model**? 🤔

Imagine that we have three classes: shark, turtle and ray.

CONFUSION MATRIX

ACTUAL\PREDICTED	SHARK	TURTLE	RAY
SHARK	50	5	10
TURTLE	3	60	2
RAY	5	4	45

Let's calculate precision and recall metrics for the shark class...

TP = 50

FP = 3+5 = 8

FN = 5+10 = 15

Precision = $TP \,/\, TP + \; FP$ = 50 / (50+ 8) = 0.86

Recall = $TP \,/\, TP + \; FP$ = 50/ (50+15) = 0.77

Now we can calculate the F1-score!

Funded by the European Union

Performance Metrics

12. F-SCORE

$$F_1 = 2\frac{precision \times recall}{(precision + recall)}$$

How it works in a **multiclass model**? 🤔

Imagine that we have three classes: shark, turtle and ray.

CONFUSION MATRIX

ACTUAL\PREDICTED	SHARK	TURTLE	RAY
SHARK	50	5	10
TURTLE	3	60	2
RAY	5	4	45

If we calculate the F1-score for each class...

CLASS	INDIVIDUAL F1-SCORE
SHARK	0.85
TURTLE	0.70
RAY	0.90

However, this table does not give us a performance indicator that allows us to compare our model against others. To do so, we require a multi-class measure of Precision and Recall.

Funded by the European Union

12. F-SCORE

- **Macro F1 score:** The macro-averaged F1 score (or macro F1 score) is computed using the arithmetic mean of all the per-class F1 scores. In the previous example it would be Macro F1 = (0.85+0.7+0.9)/3 = 0.82

$$Macro\ F1 = \frac{\sum_{k=1}^{K} F1_k}{K}$$

- **Weighted Averaged F1 score:** The weighted-averaged F1 score is calculated by taking the mean of all per-class F1 scores while considering each class's support.

Funded by
the European Union

12. F-SCORE

- **Micro F1** (if you are interested in general performance)

$$Micro\ Average\ Precision\ = \frac{\sum_{k=1}^{K} TP_k}{Grand\ Total}$$

$$Micro\ Average\ Recall\ = \frac{\sum_{k=1}^{K} TP_k}{Grand\ Total}$$

Long story short, we may see that Micro-Average Precision and Recall are just the same values, therefore the MicroAverage F1-Score is just the same as well (the harmonic mean of two equal values is just the value).

$$Micro\ AverageF1\ = \frac{\sum_{k=1}^{K} TP_k}{Grand\ Total}$$

Funded by
the European Union

13. Intersection over Union (IoU)

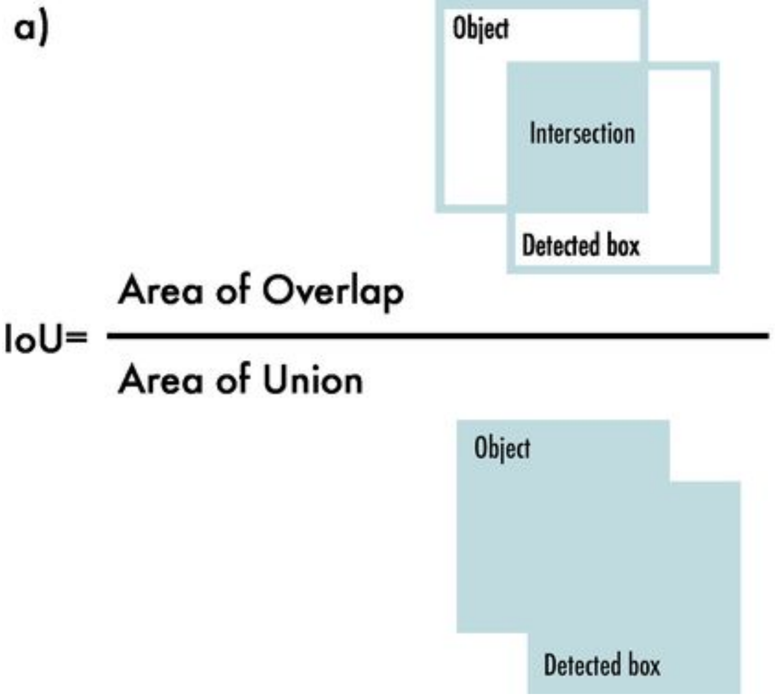
At the moment, we have only spoke about **classification metrics**. But if we are working with models that **detect and localize objects**, we need something more: evaluate if we detect correctly each object, in their correct place and with how much precision.

Funded by the European Union

Performance Metrics

13. Intersection over Union (IoU)

At the moment, we have only spoke about **classification metrics**. But if we are working with models that **detect and localize objects**, we need something more: evaluate if we detect correctly each object, in their correct place and with how much precision.



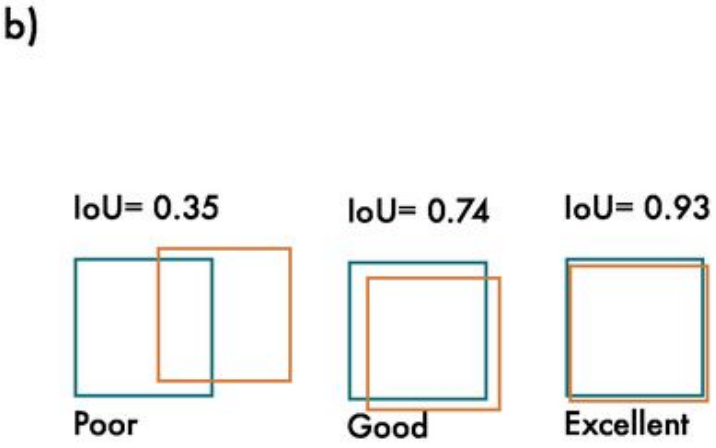


Image source:
A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond – Scientific Figure on ResearchGate. Available at: https://www.researchgate.net/figure/ntersection-over-Union-IoU-a-The-IoU-is-calculated-by-dividing-the-intersection-of_fig1_369760111 [Accessed 14 Apr 2025].

Funded by
the European Union

Performance Metrics

14. P-R curve

- Set a **threshold** to decide if a prediction is correct.
- EXAMPLE : If IoU > 0.5 → **True Positive (TP)**
  Else → **False Positive (FP)**
  The real boxes that are not detected are **False Negativ**

The Precision-Recall curve (P-R Curve):

- Different IoU thresholds → different TP, FP, FN counts
- This gives us different points on the **Precision-Recall (P-R) Curve**
- Lower IoU threshold → higher recall, lower precision
- Higher IoU threshold → higher precision, lower recall

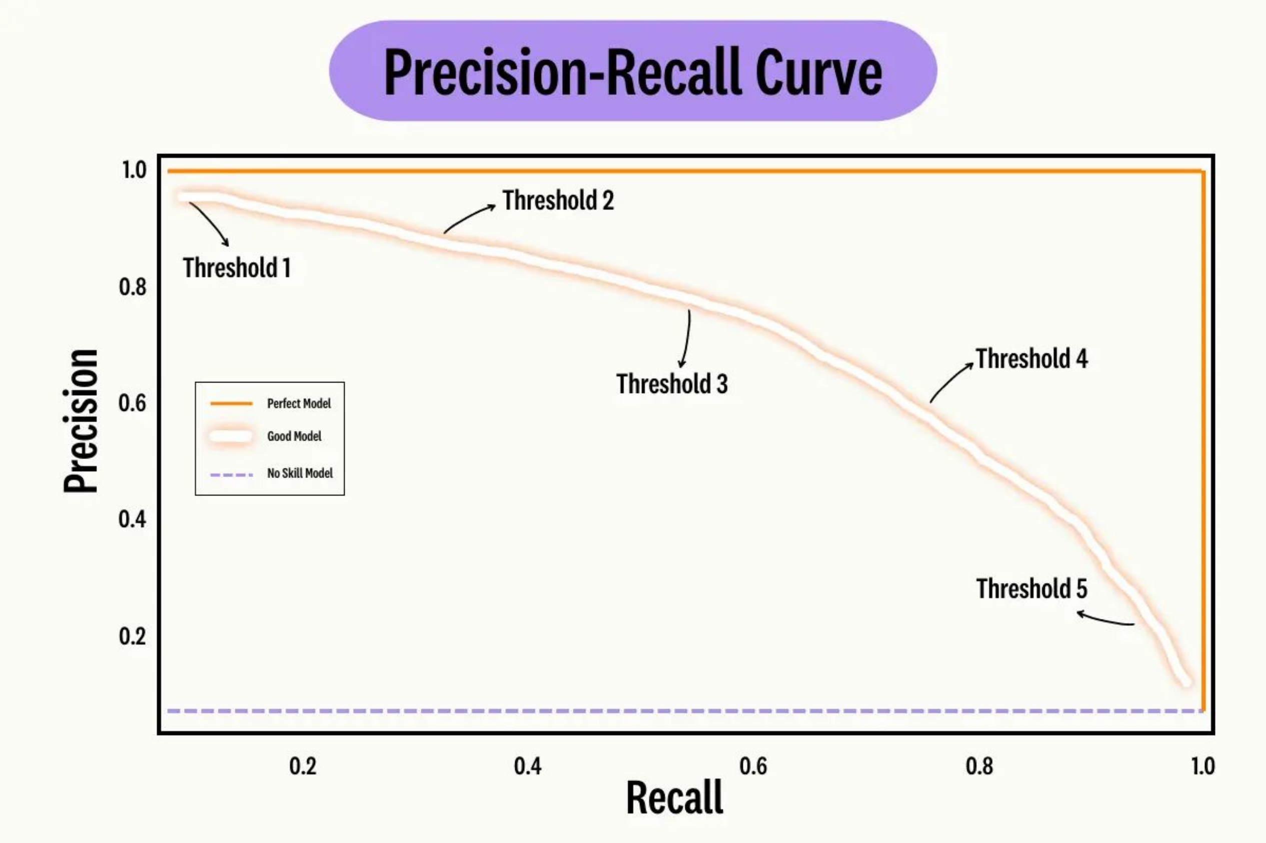
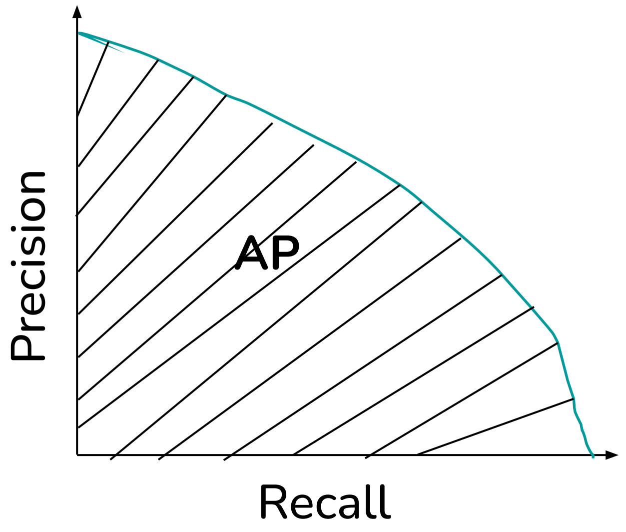


Image source: Kili Technology – "Mean Average Precision (mAP): A Complete Guide." Available at: [*https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide*](https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide) *[Accessed 14 Apr 2025].*

Funded by the European Union

15. Average Precision (AP)

We can calculate AP as a way to summarize the precision-recall curve into a single value representing the average of all precisions. **AP is the area behind the curve**, the bigger is the area, the best is the model performance.

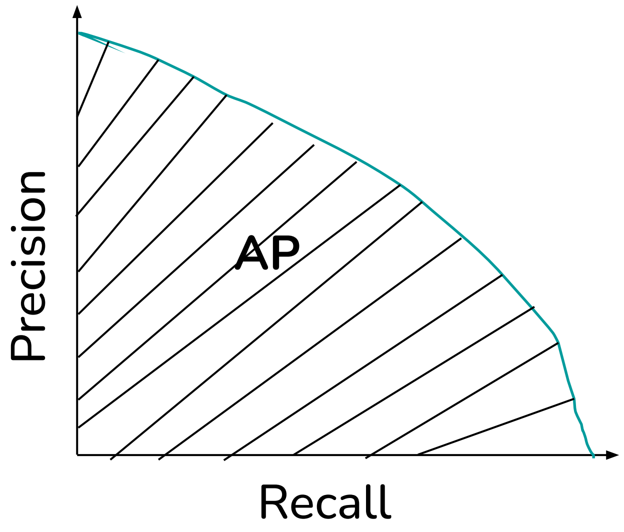


Funded by
the European Union

Performance Metrics

15. Average Precision (AP)

We can calculate AP as a way to summarize the precision-recall curve into a single value representing the average of all precisions. **AP is the area behind the curve**, the bigger is the area, the best is the model performance.



Average Precision (AP) is calculated **separately for each class**, which is essential when evaluating multi-class models. To summarize the overall performance, we compute the **Mean Average Precision (mAP)** across all classes

Funded by the European Union

16. Mean Average Precision (mAP)

Final evaluation metric

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

Where:
N = number of classes
AP_i = Average Precision for i classes

Why is mAP important?

- mAP allows **fair comparison** between models on multi-class datasets.
- It captures both **precision and recall**, providing a **balanced view** of performance.
- Widely used in object detection benchmarks (e.g. **COCO**, **Pascal VOC**).

The value of **mAP ranges from 0 to 1**:

- **1.0 (or 100%)** means perfect detection across all classes.
- **0.0** means the model failed to detect objects correctly.

The higher the mAP, the better the model performs across all classes

Funded by the European Union

17. SUMMARY

📦 **Classification**: Accuracy, Precision, Recall, F1-score

🧭 **Object Detection & Segmentation**: IoU, AP, mAP

Concept	What it tell us
IoU	Evaluates **how accurate the predicted bounding boxes** are
P-R Curve	Shows **how precision and recall vary** as the confidence threshold changes
AP	Measures the overall prediction quality for a single class
mAP	Measures the global performance, averaging the APs across all classes

Funded by the European Union

Practice

colab

https://colab.research.google.com/drive/1ckAZMpABRG7NDXLX1clVZEXbx-cYVfMY

Funded by
the European Union