Institución Universitaria
Reacreditada en Alta Calidad
80 Años

Supervised Learning: Regression

Outline

1. Training, Validation and Testing phases
2. Regression Definition
3. Error measurements
4. Overtraining, Undertraining and Regularization

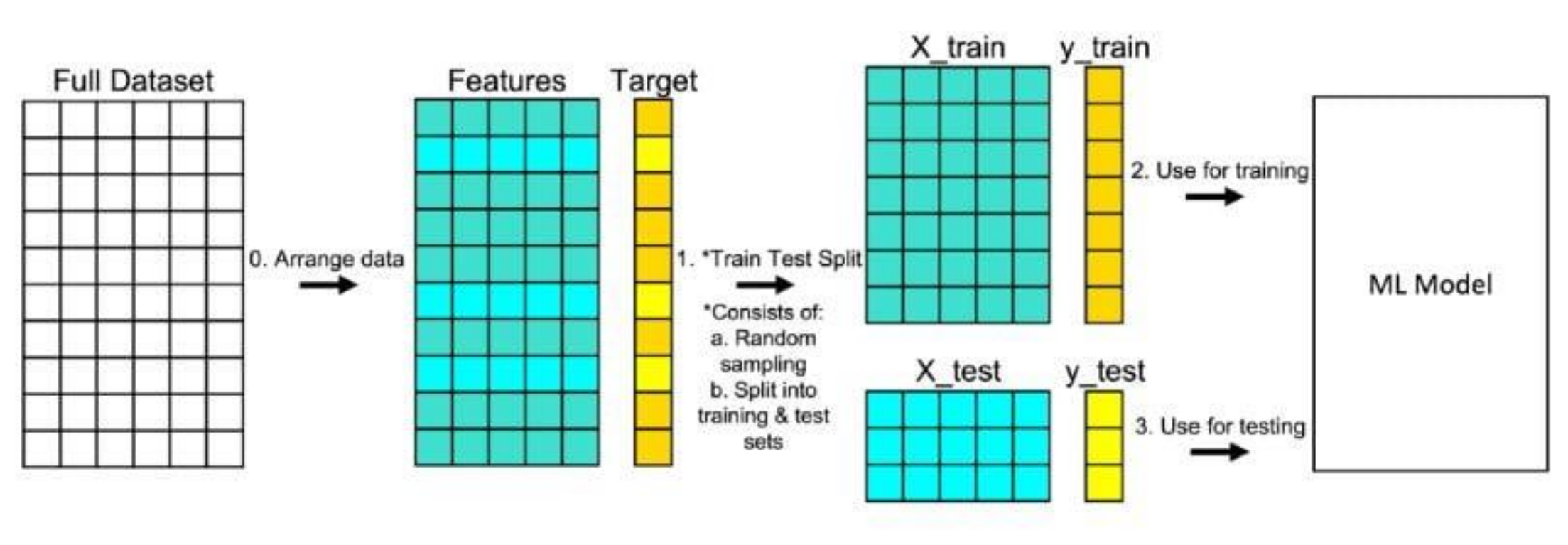
Training, Validation and Testing phases

Train test split is a model validation procedure that allows you to simulate how a model would perform on new/unseen data.

Hyperparameter tuning: model's parameters are tuned according to an optimizacion task during the training phase.

Inference (Testing): After finding a good set of hyperparameters, the trained model is tested on unseen data, in order to check its performance.

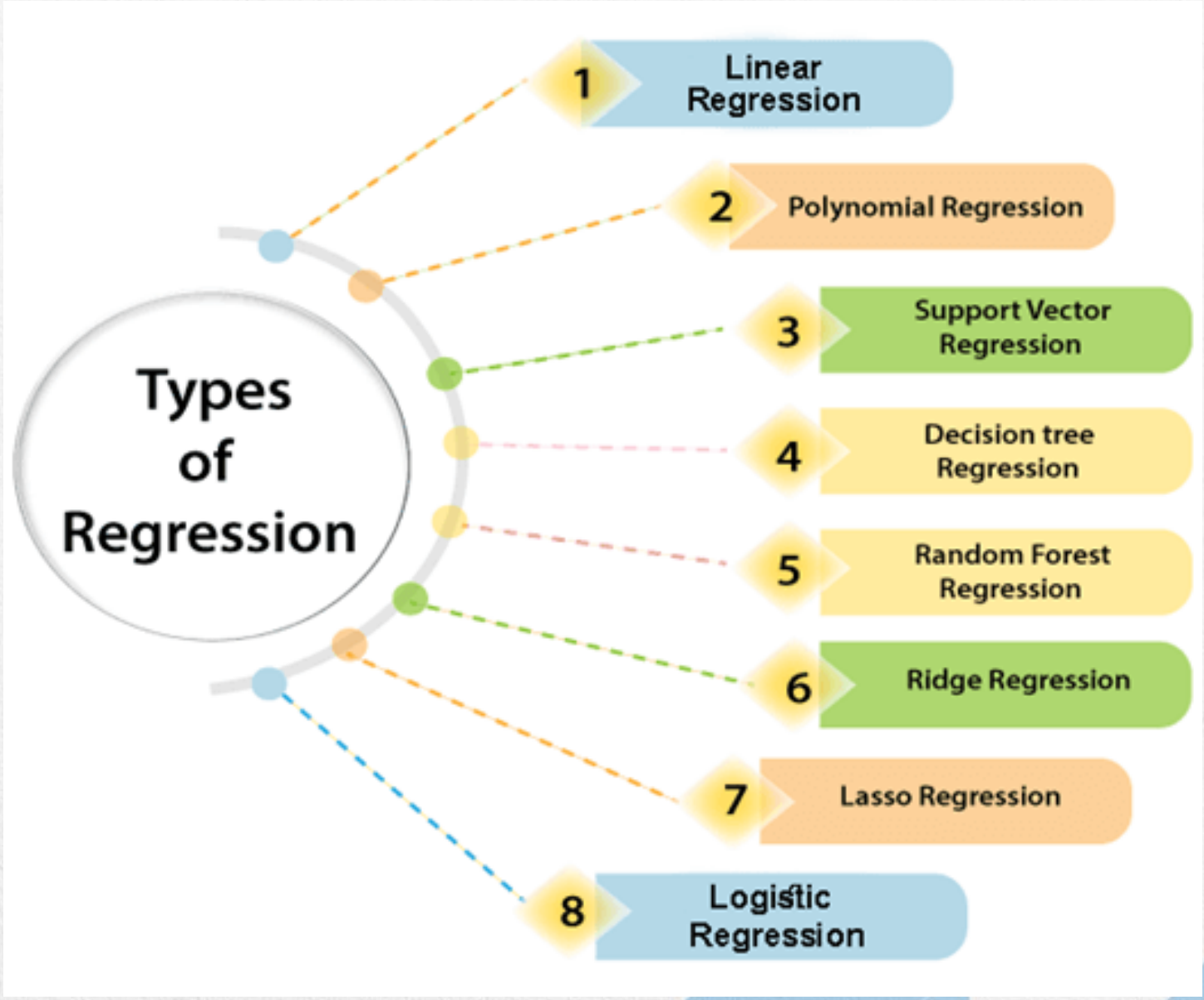
Training, Validation and Testing phases



Example of splitting the data into Training and Test sets.

https://builtin.com/data-science/train-test-split

Regression analysis



Outline

1. Training, Validation and Testing phases
2. Regression Definition
3. Error measurements
4. Overtraining, Undertraining and Regularization

Regression analysis

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables.

Simple regression

$$f(x) = y$$

Multiple regression

$$f(x_1, x_2, \dots, x_n) = y$$

$$f(\boldsymbol{x}_i) = y_i$$

Outline

1. Training, Validation and Testing phases
2. Regression Definition
3. Error measurements
4. Overtraining, Undertraining and Regularization

Error measurements

Most typical objective functions for regression are, the root mean squared error (RMSE) and the mean absolute error (MAE).

RMSE

$$f(x_i, \theta\ ) = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i(x_i, \theta) - y_i(x_i))^2$$

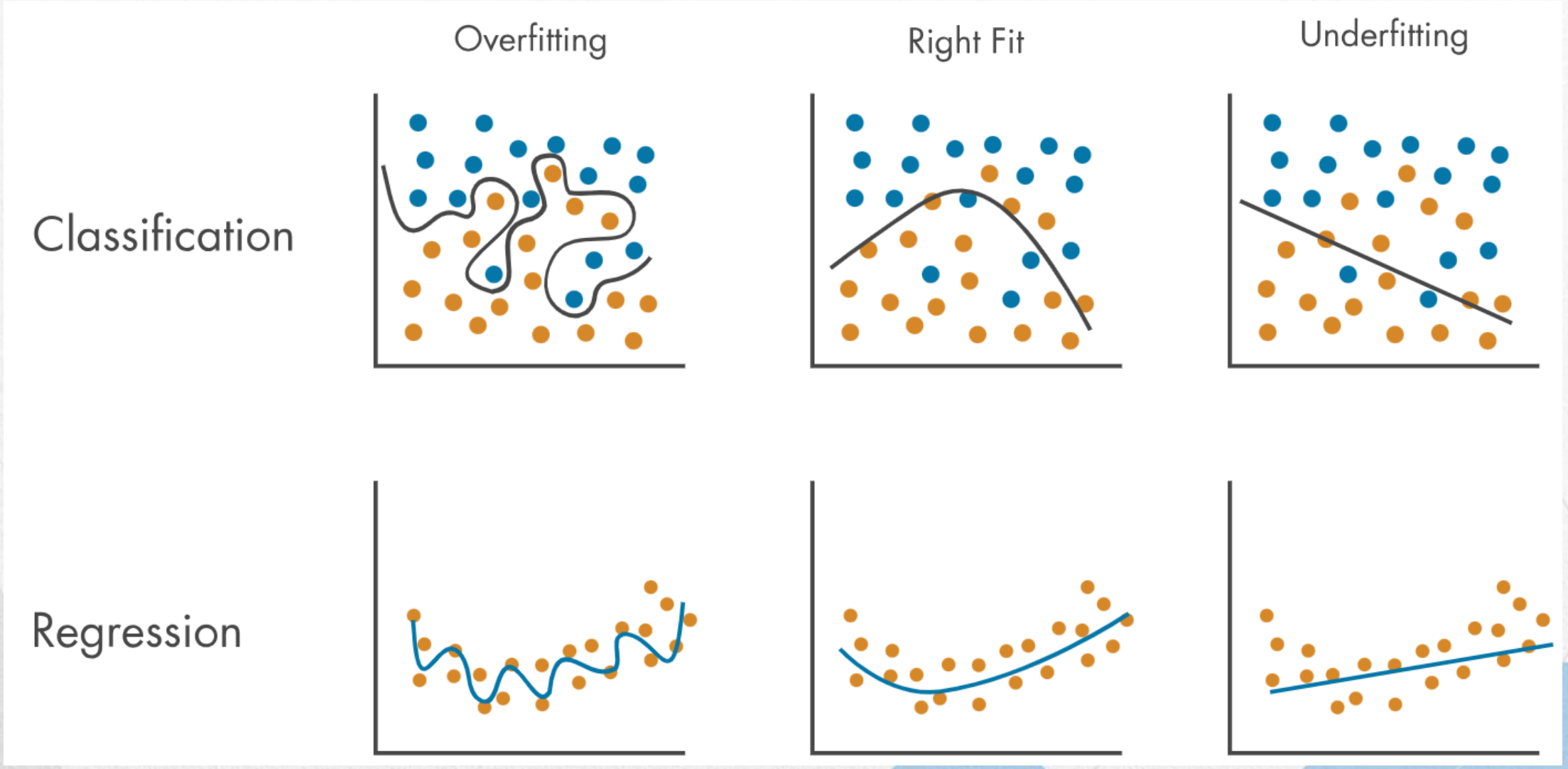
MAE

$$f(\theta) = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y}_i(\theta) - y_i|$$

Outline

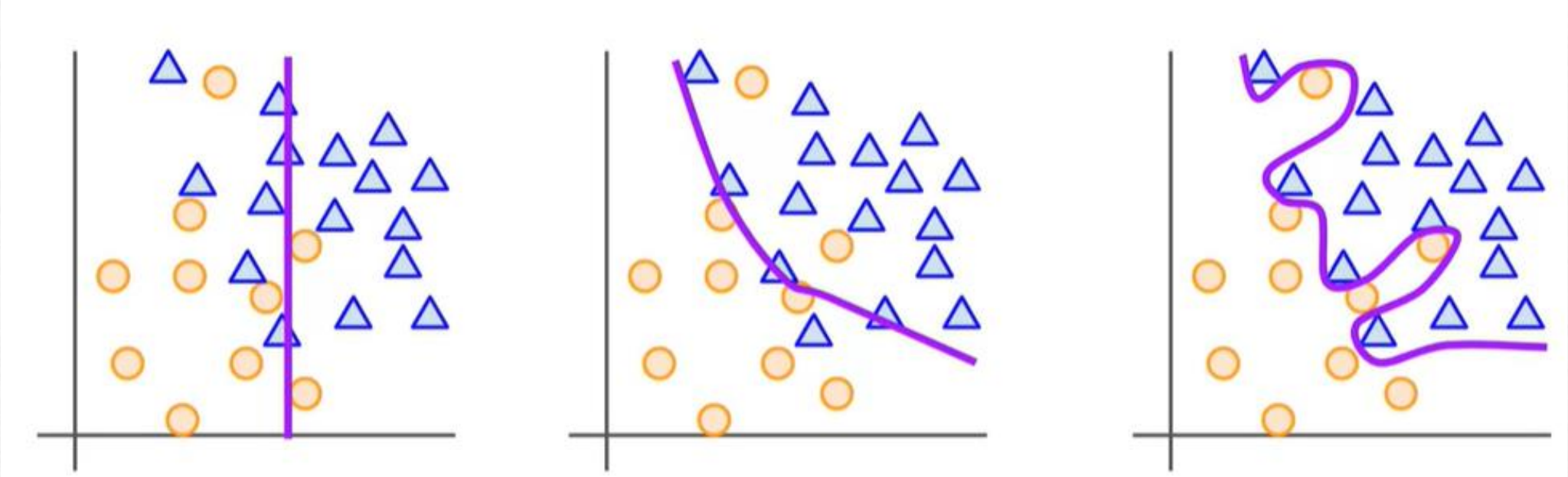
1. Training, Validation and Testing phases
2. Regression Definition
3. Error measurements
4. Overtraining, Undertraining and Regularization

Overfitting



https://www.mathworks.com/discovery/overfitting.html

Overfitting



Underfit: Model fails to capture trends in the data

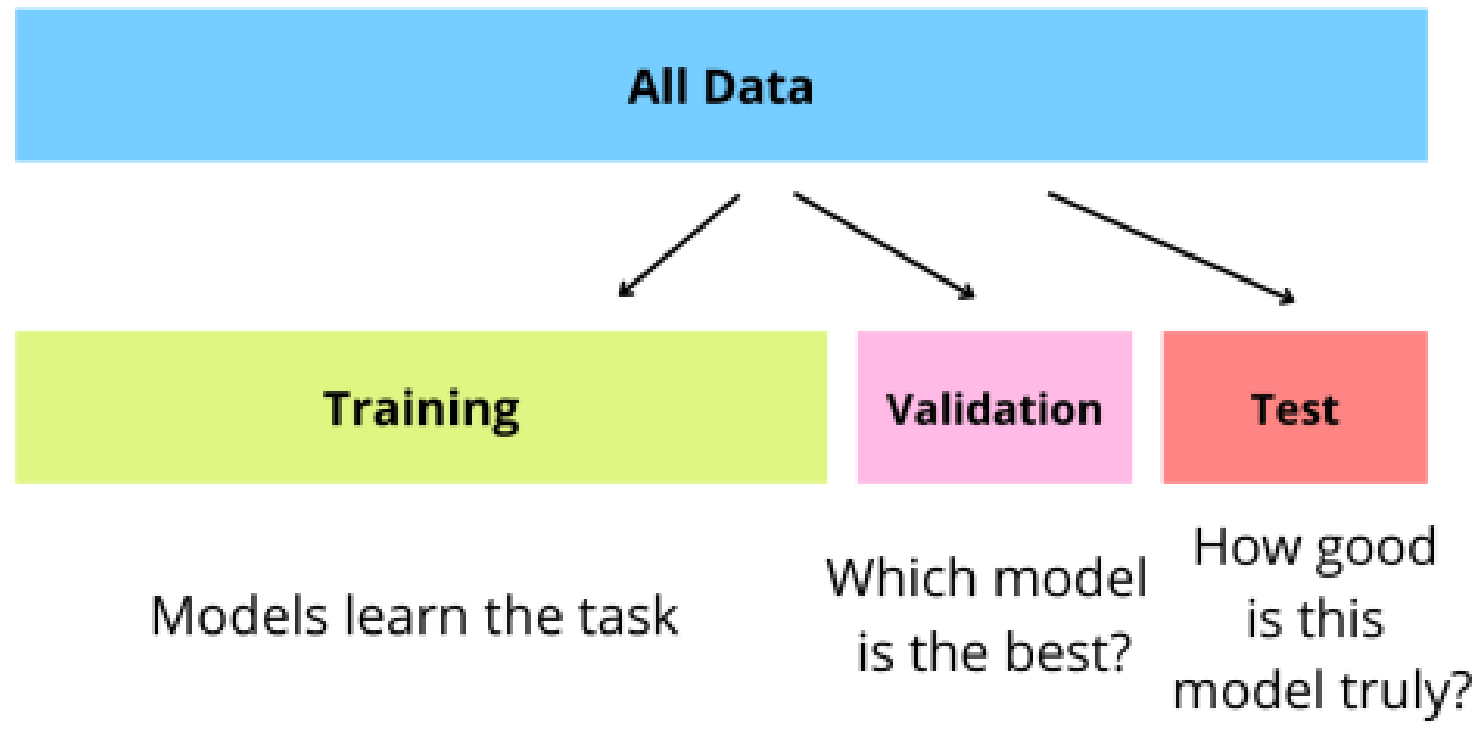
Good fit: Model captures trends and can generalize to unseen data

Overfit: Model captures training data trends but fails on unseen data

https://www.coursera.org/learn/introduction-to-embedded-machine-learning/

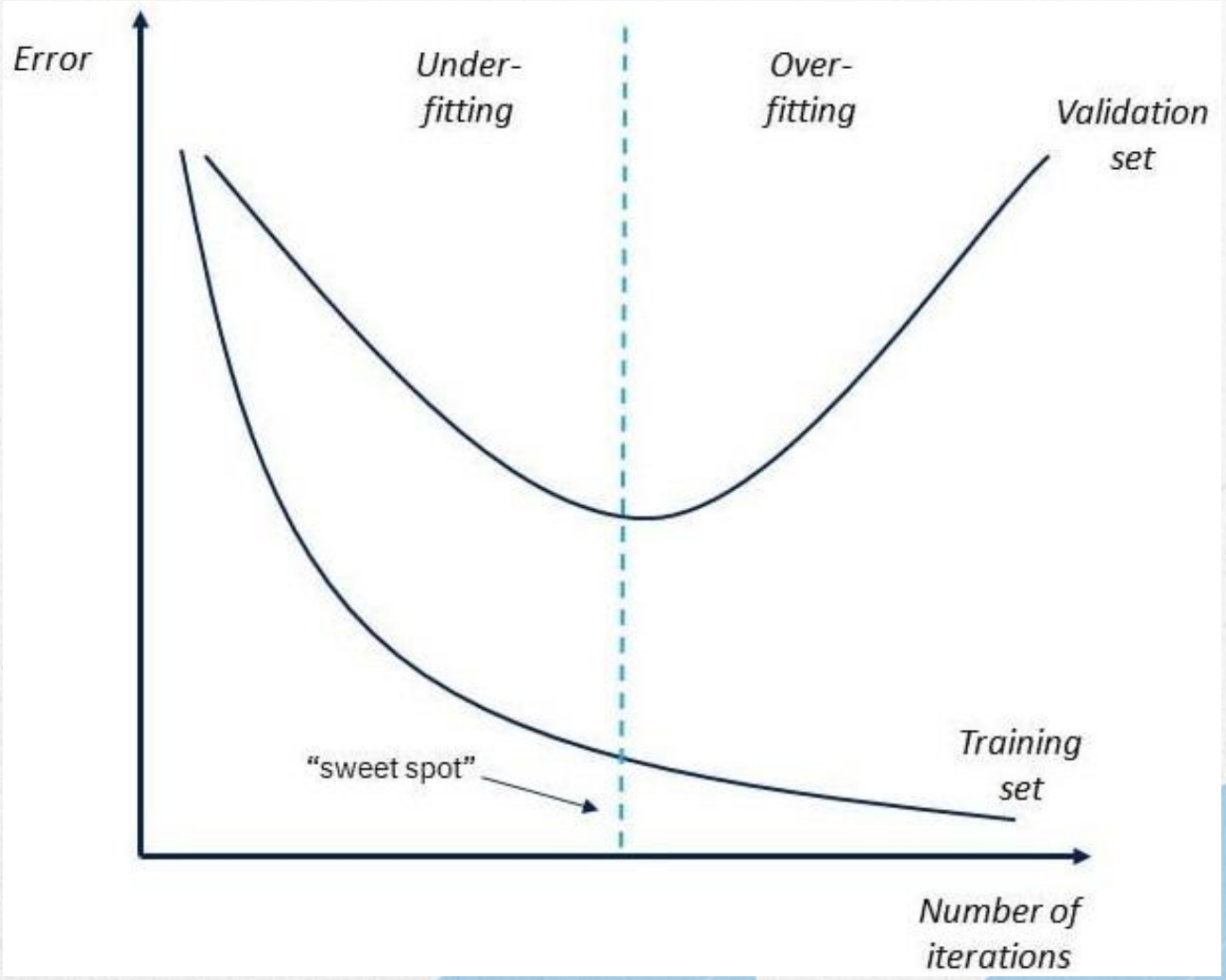
Training in Deep Learning

Validation data is used during training phase to avoid overfitting and perform early stopping.



Training in Deep Learning

Validation data is used during training phase to avoid overfitting and perform early stopping.



Classification error measurements

Let's consider a scenario where we have built a spam detection model using a dataset of 1,000 emails, with 800 legitimate emails and 200 spam emails. After training and testing the model, we obtained the following results:

Confusion Matrix:

	Predicted: Spam	Predicted: Legitimate
Actual: Spam	160	10
Actual: Legitimate	20	810

https://www.linkedin.com/pulse/understanding-performance-metrics-machine-learning-precision-saini/

Precision

Precision measures the accuracy of positive predictions made by the model. In our case, it represents the proportion of correctly predicted spam emails out of all emails predicted as spam. Using the confusion matrix, we can calculate precision as:

Precision = True Positives / (True Positives + False Positives)
= 160 / (160 + 20) = 0.8889

Recall

Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify positive instances. It represents the proportion of correctly predicted spam emails out of all actual spam emails. Using the confusion matrix, we can calculate recall as:

Recall = True Positives / (True Positives + False Negatives)
= 160 / (160 + 10) = 0.9412

F1-score

The F1 score combines precision and recall into a single metric. It provides a balanced measure that considers both false positives and false negatives. The F1 score can be calculated as the harmonic mean of precision and recall:

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
= 2 * (0.8889 * 0.9412) / (0.8889 + 0.9412) = 0.9143

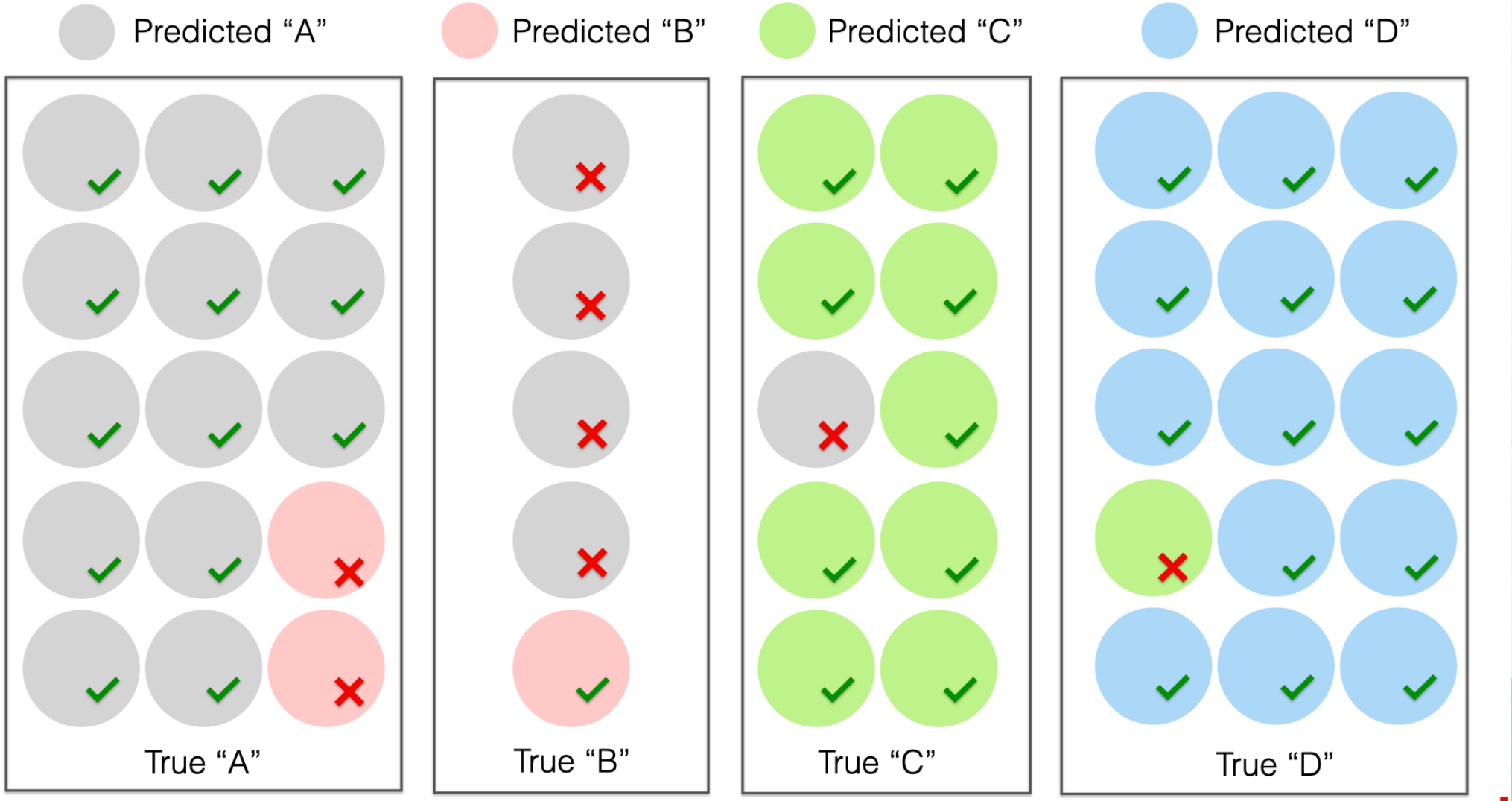
Multiclass

Accuracy measures the proportion of correctly classified cases from the total number of objects in the dataset. To compute the metric, divide the number of correct predictions by the total number of predictions made by the model.

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

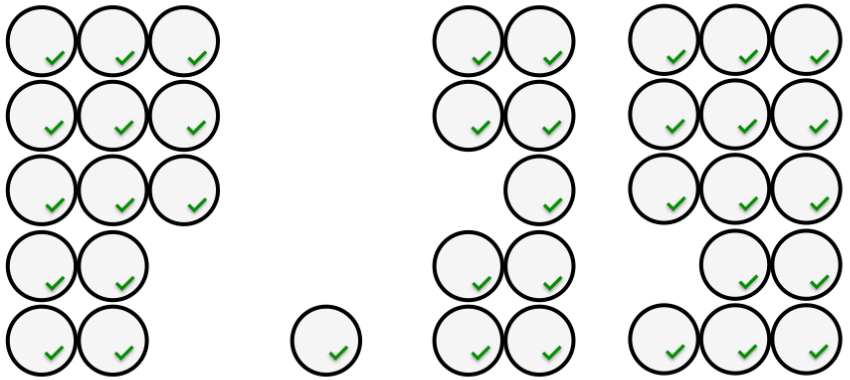
https://www.evidentlyai.com/classification-metrics/multi-class-metrics

Multiclass



Multiclass

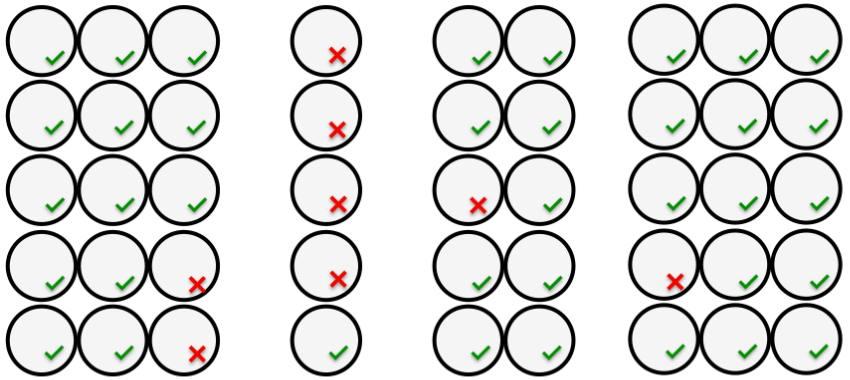
correct predictions



37

Accuracy =

all predictions



45

Multiclass

Say, you are dealing with manufacturing defect prediction. For every new product on a manufacturing line, you assign one of the categories: “no defect,” “minor defect,” “major defect,” or “scrap.” You are most interested in finding defects: the goal is to proactively inspect and take faulty products off the line.

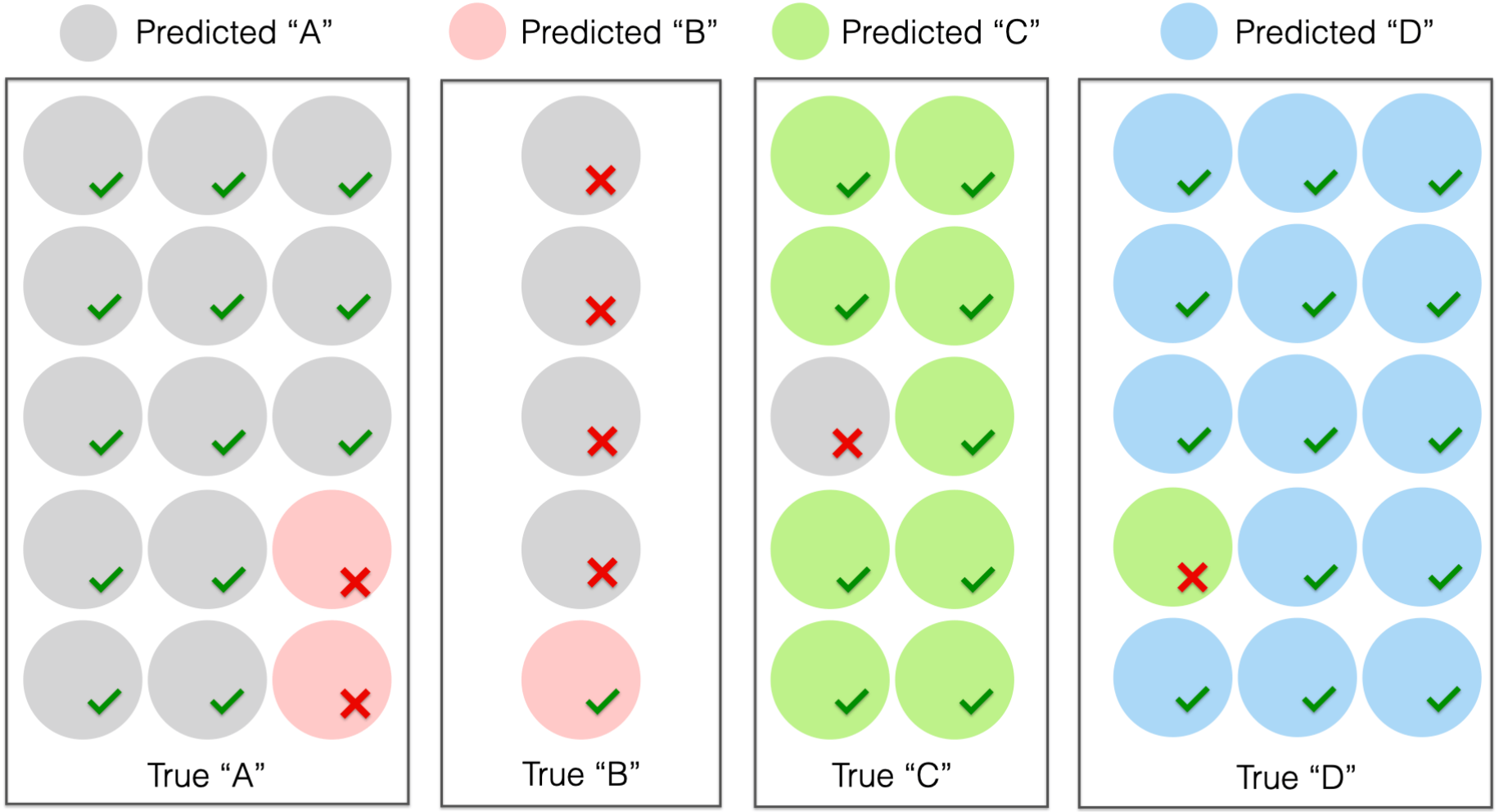
The model might be mostly correct in assigning the “no defect” and “scrap” labels but be unreasonable when predicting actual defects. Meanwhile, the accuracy metric might be high due to the successful performance of the majority classes. In this case, accuracy might not be a suitable metric.

Multiclass

$$\text{Precision}_{\text{Class A}} = \frac{TP_{\text{Class A}}}{TP_{\text{Class A}} + FP_{\text{Class A}}}$$

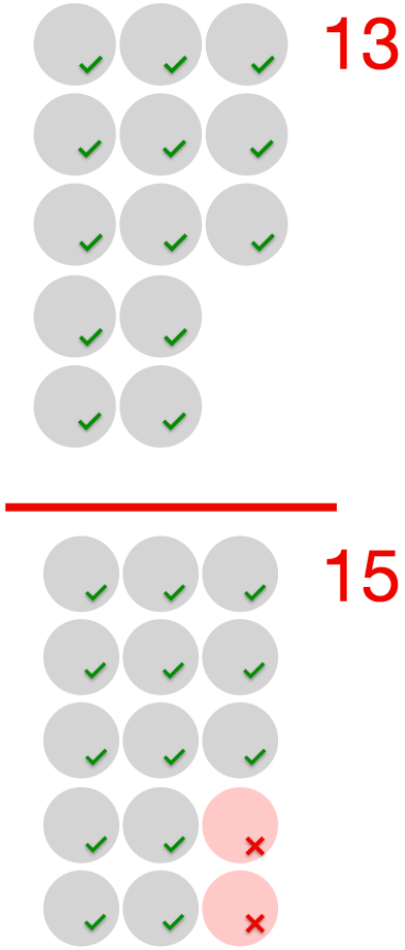
$$\text{Recall}_{\text{Class A}} = \frac{TP_{\text{Class A}}}{TP_{\text{Class A}} + FN_{\text{Class A}}}$$

Multiclass

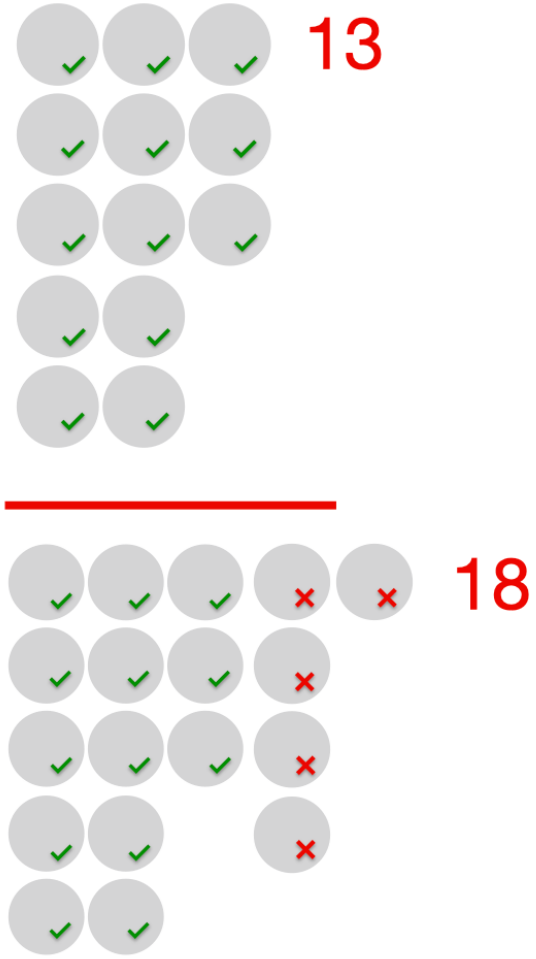


Multiclass

$$\text{Recall} = \frac{\text{correct } \textbf{Class A} \text{ predictions}}{\text{all } \textbf{Class A} \text{ Instances}}$$



$$\text{Precision} = \frac{\text{correct } \textbf{Class A} \text{ predictions}}{\text{all } \textbf{Class A} \text{ predictions}}$$



Institución Universitaria
Reacreditada en Alta Calidad
80 Años

¡MUCHAS GRACIAS!

VIGILADA MINEDUCACIÓN

www.itm.edu.co

Alcaldía de Medellín
Distrito de
Ciencia, Tecnología e Innovación