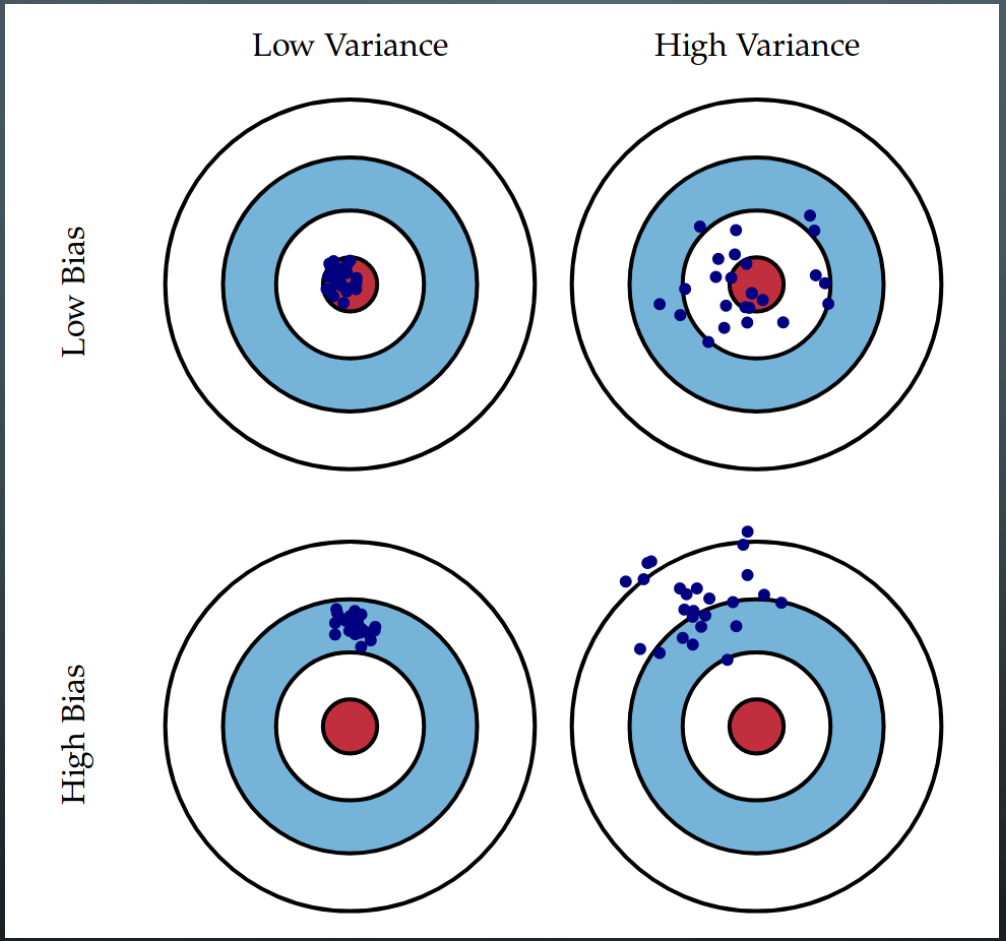
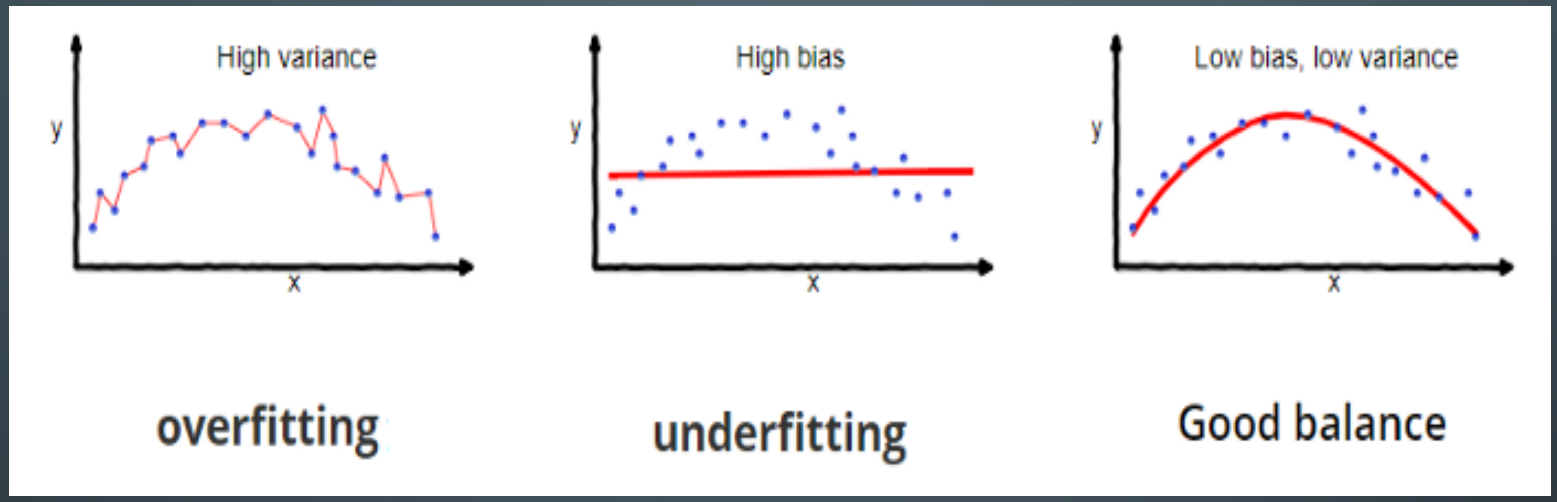
EVALUATION AND TUNING

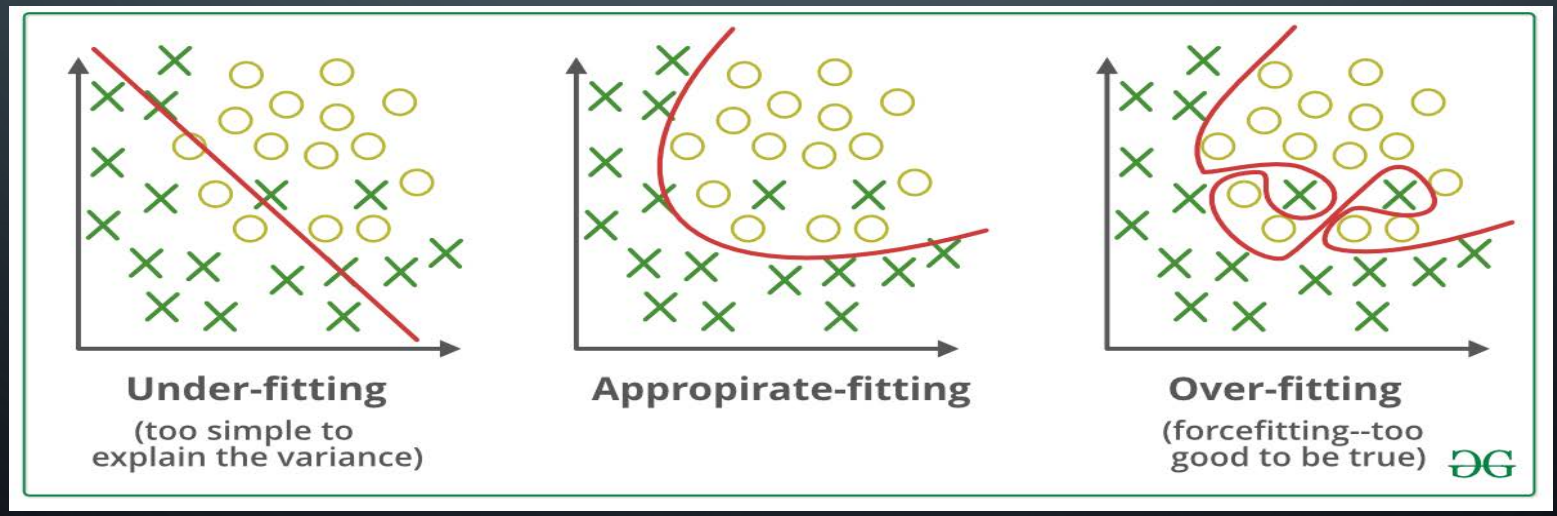
MOHAMMAD GHODDOSI

BIAS AND VARIANCE

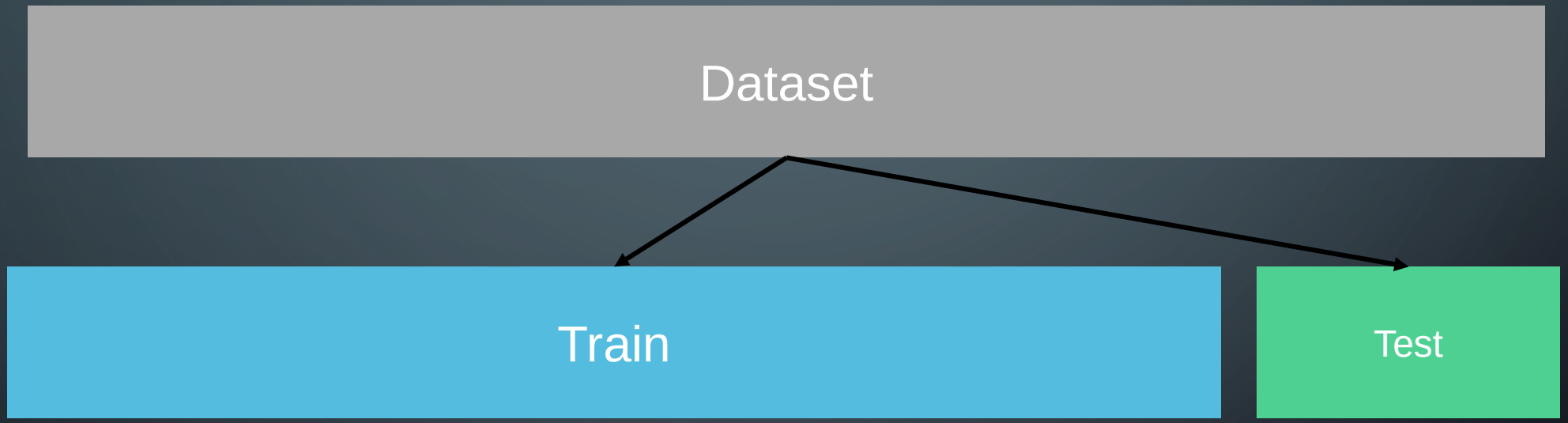


OVERFITTING AND UNDERFITTING

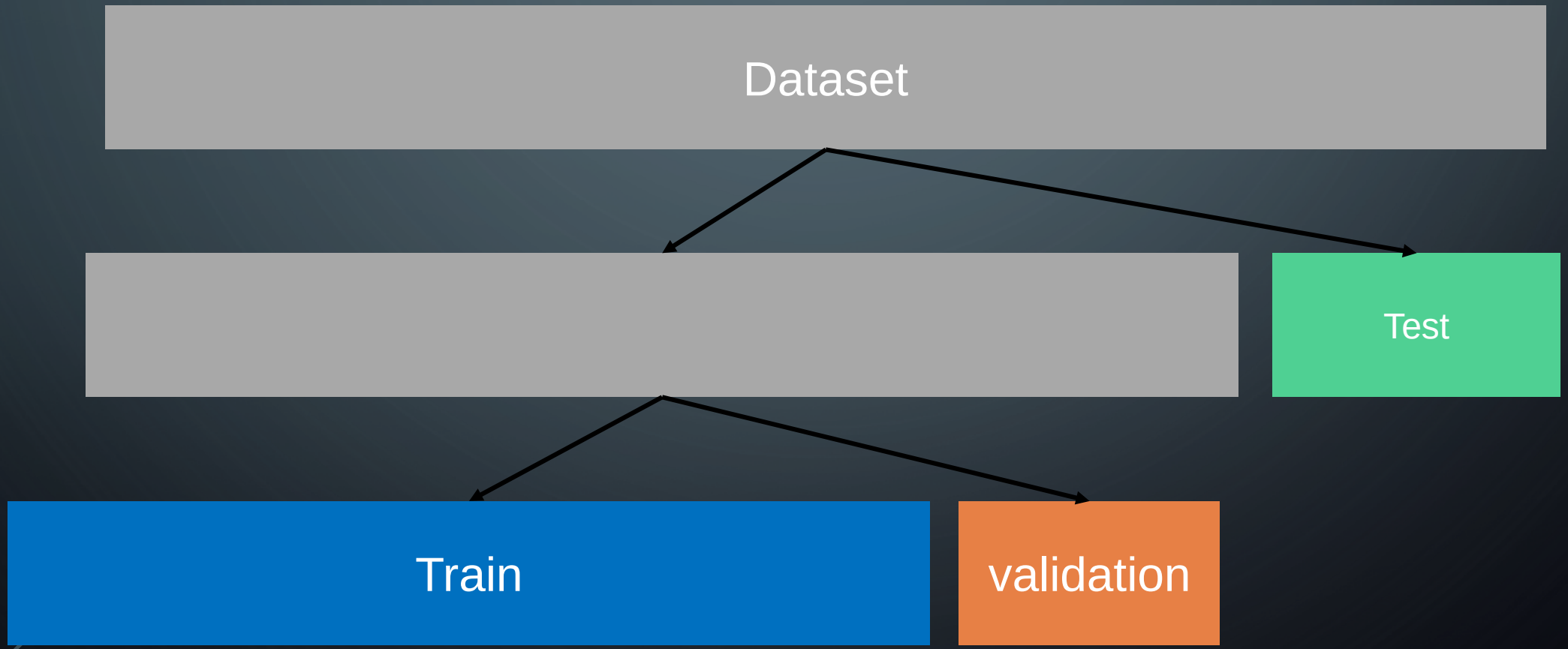




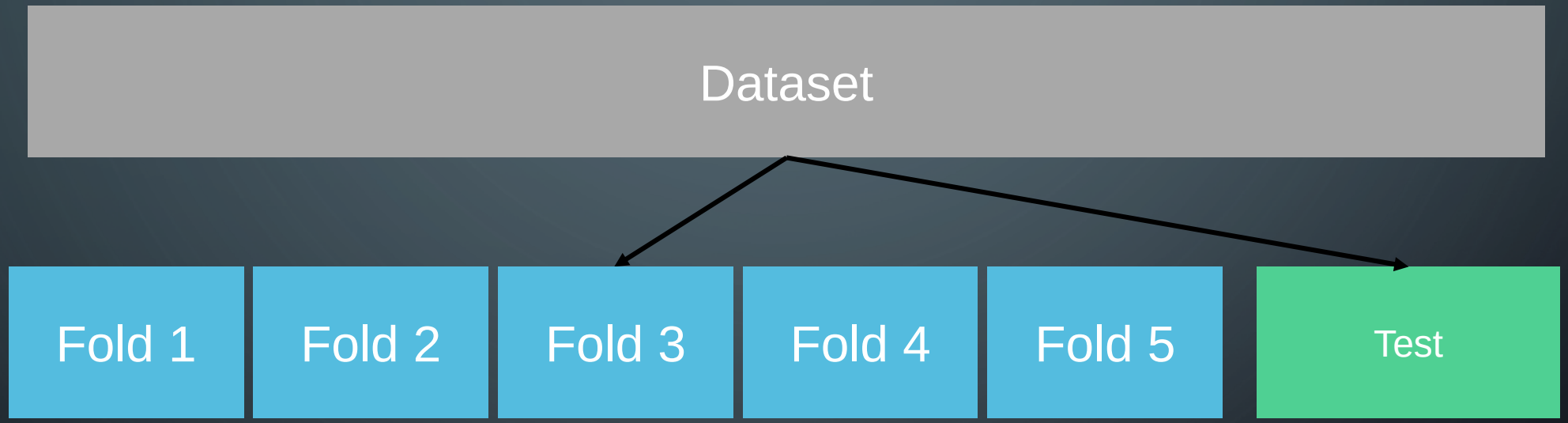
TRAIN-TEST SPLIT



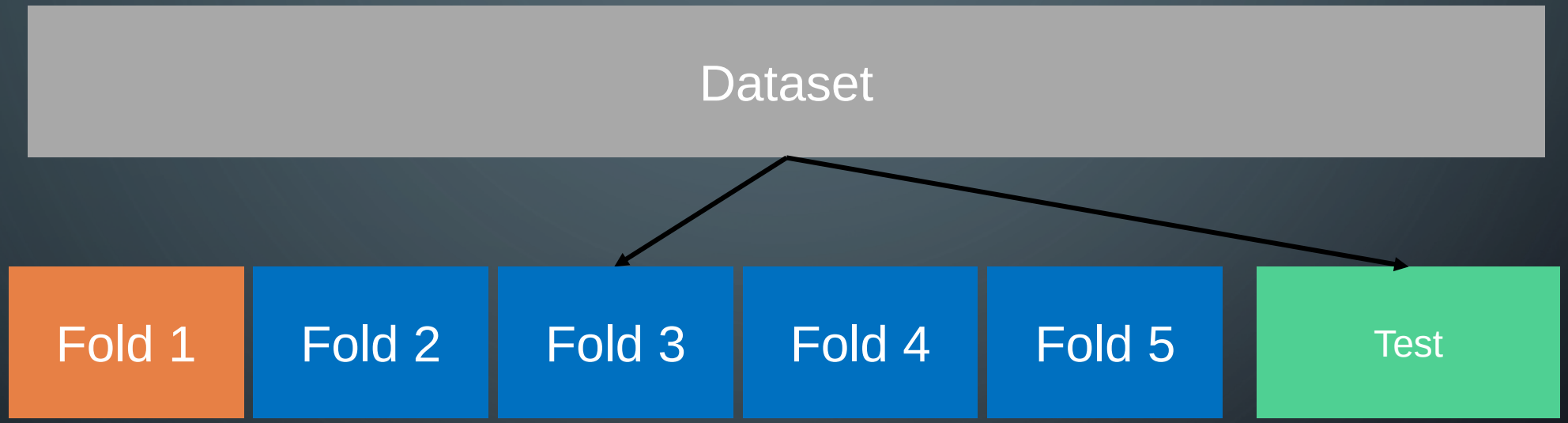
TRAIN-TEST - VALIDATION SPLIT



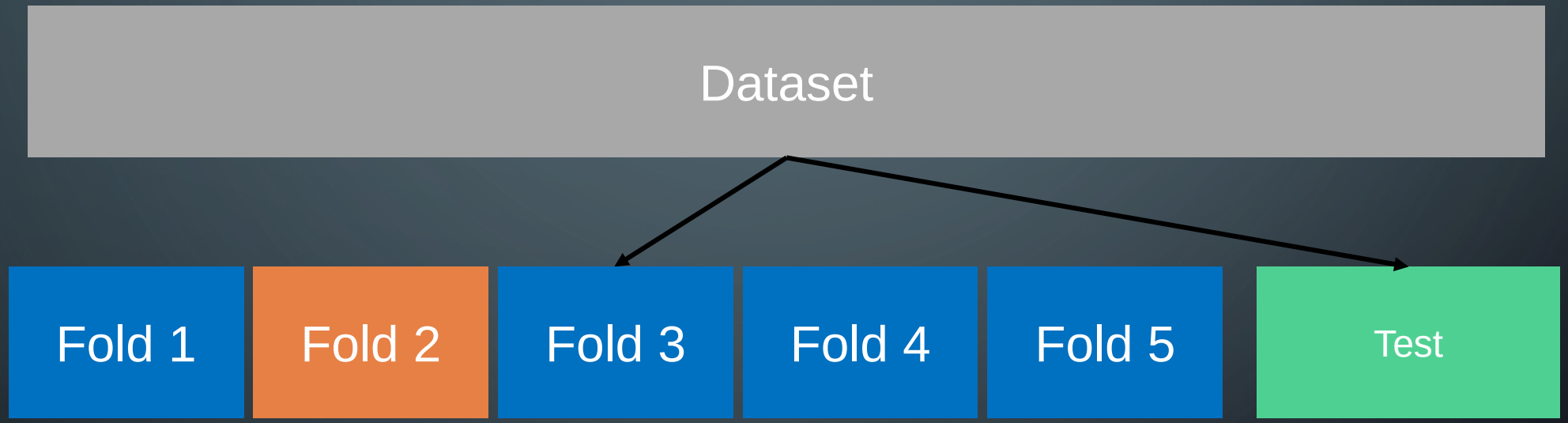
K-FOLD CROSS VALIDATION



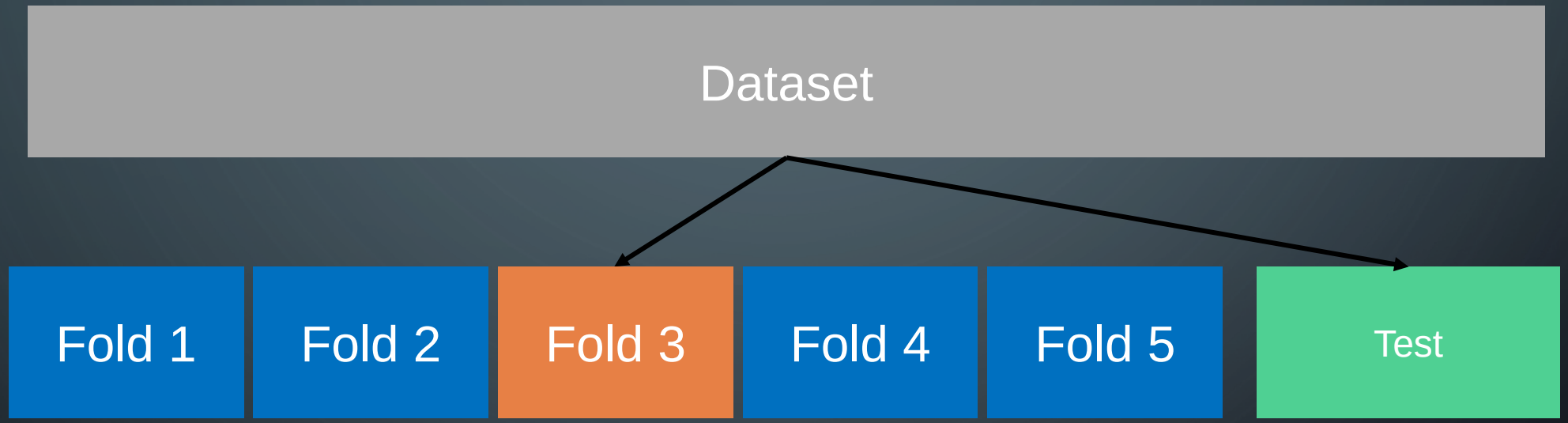
K-FOLD CROSS VALIDATION



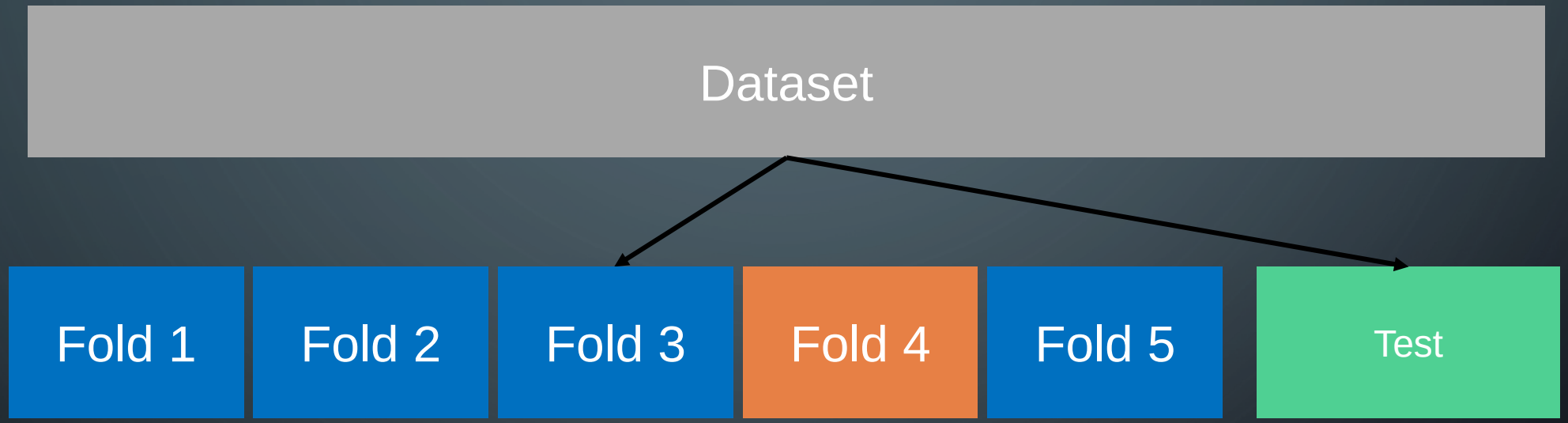
K-FOLD CROSS VALIDATION



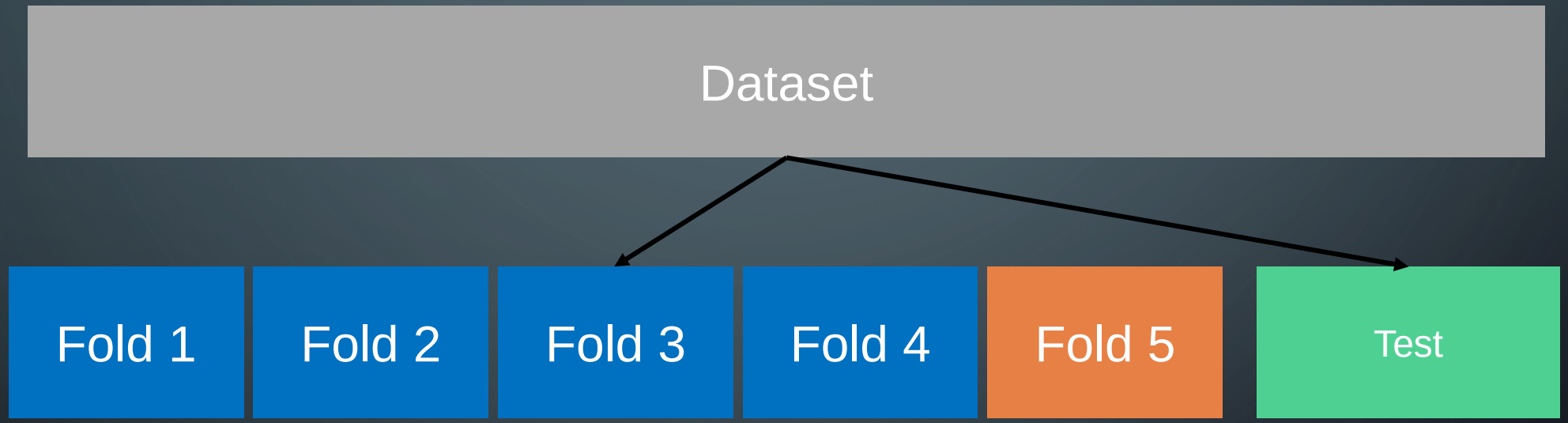
K-FOLD CROSS VALIDATION



K-FOLD CROSS VALIDATION



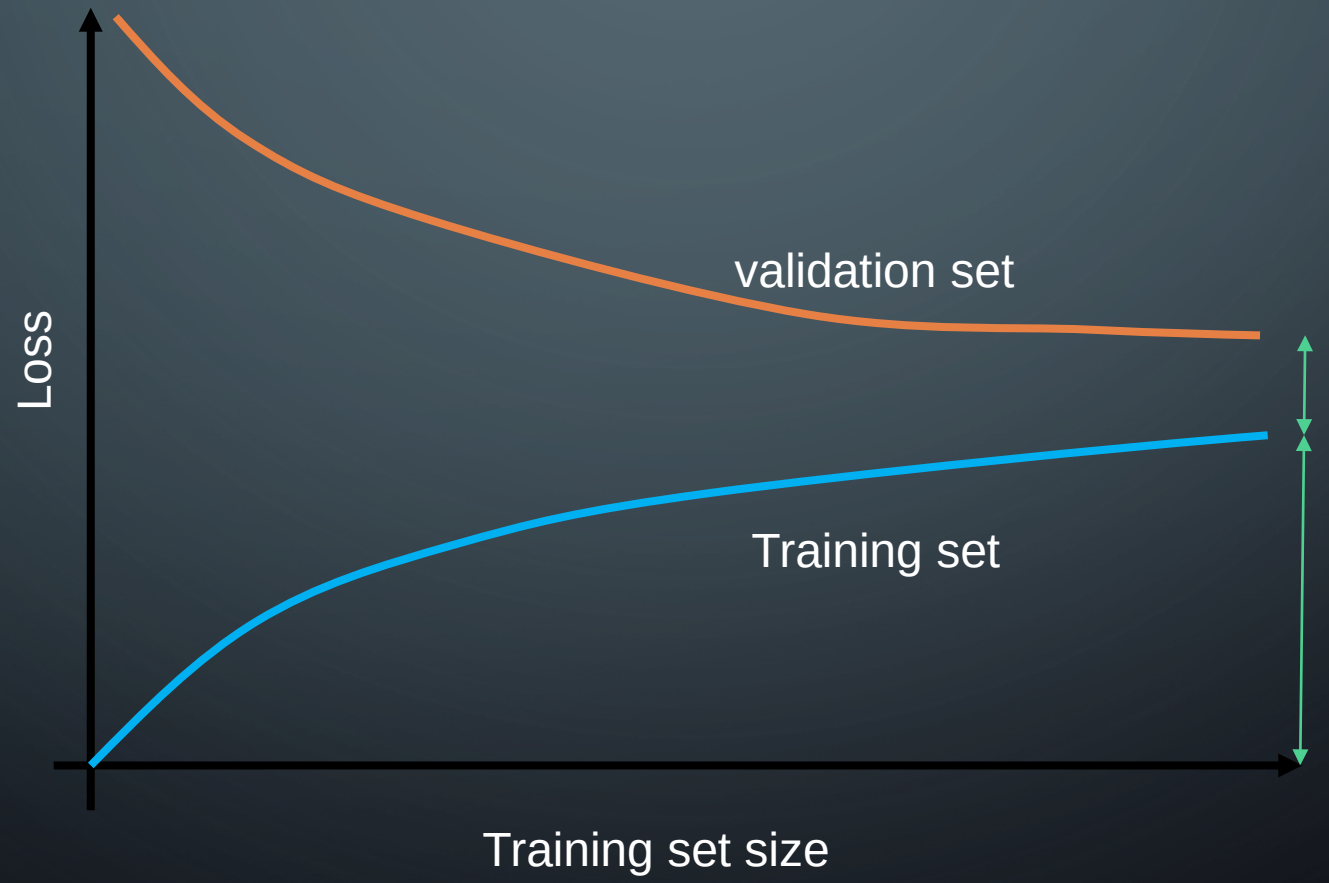
K-FOLD CROSS VALIDATION



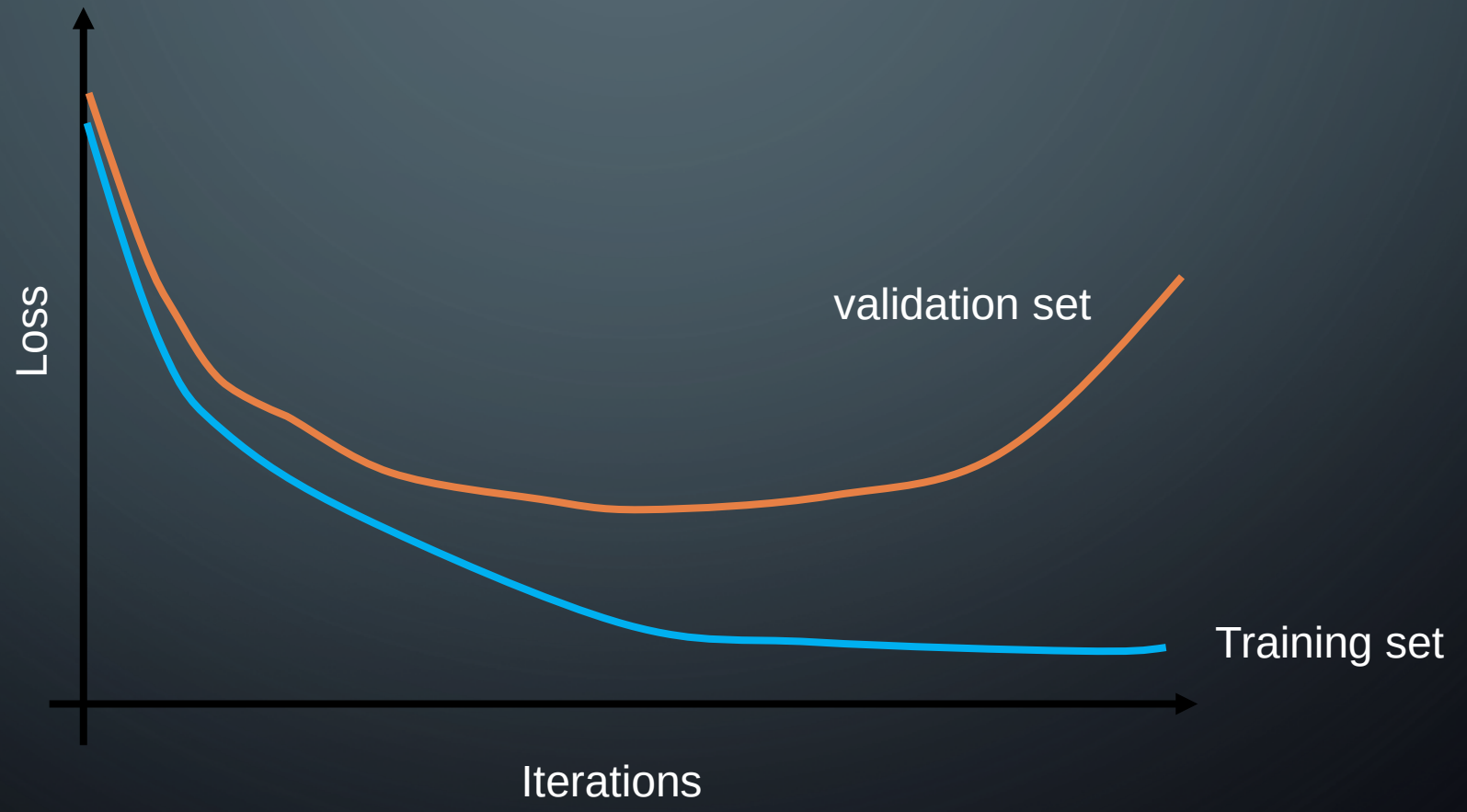
HYPER PARAMETER TUNING

- Grid search
- Random search
- Evolutionary optimization
- …

LEARNING CURVE (OLD)



LEARNING CURVE (NEW)



HANDLING UNDERFITTING

- Increase model capacity
  - Add polynomial features
- Increase features
  - Get more features
  - Use feature extraction
- Reduce regularization
- Use more powerful models
- …

HANDLING OVERFITTING

- Reduce model capacity
  - Reduce parameters
  - Reduce polynomial degree
  - Remove some features
- Apply regularization
- Increase data size
- Early stopping
- …

PERFORMANCE MEASURES

- Accuracy
- Confusion matrix
- Precision
- Recall
- F1-score
- ROC-AUC

PERFORMANCE MEASURES (ACC)

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

$$Accuracy(ACC) = \frac{\#TP + \#TN}{\#Positive + \#Negative}$$

PERFORMANCE MEASURES (PRECISION)

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

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

PERFORMANCE MEASURES (RECALL)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$recall(TPR) = \frac{\#TP}{\#TP + \#FN}$$

PERFORMANCE MEASURES (F1-SCORE)

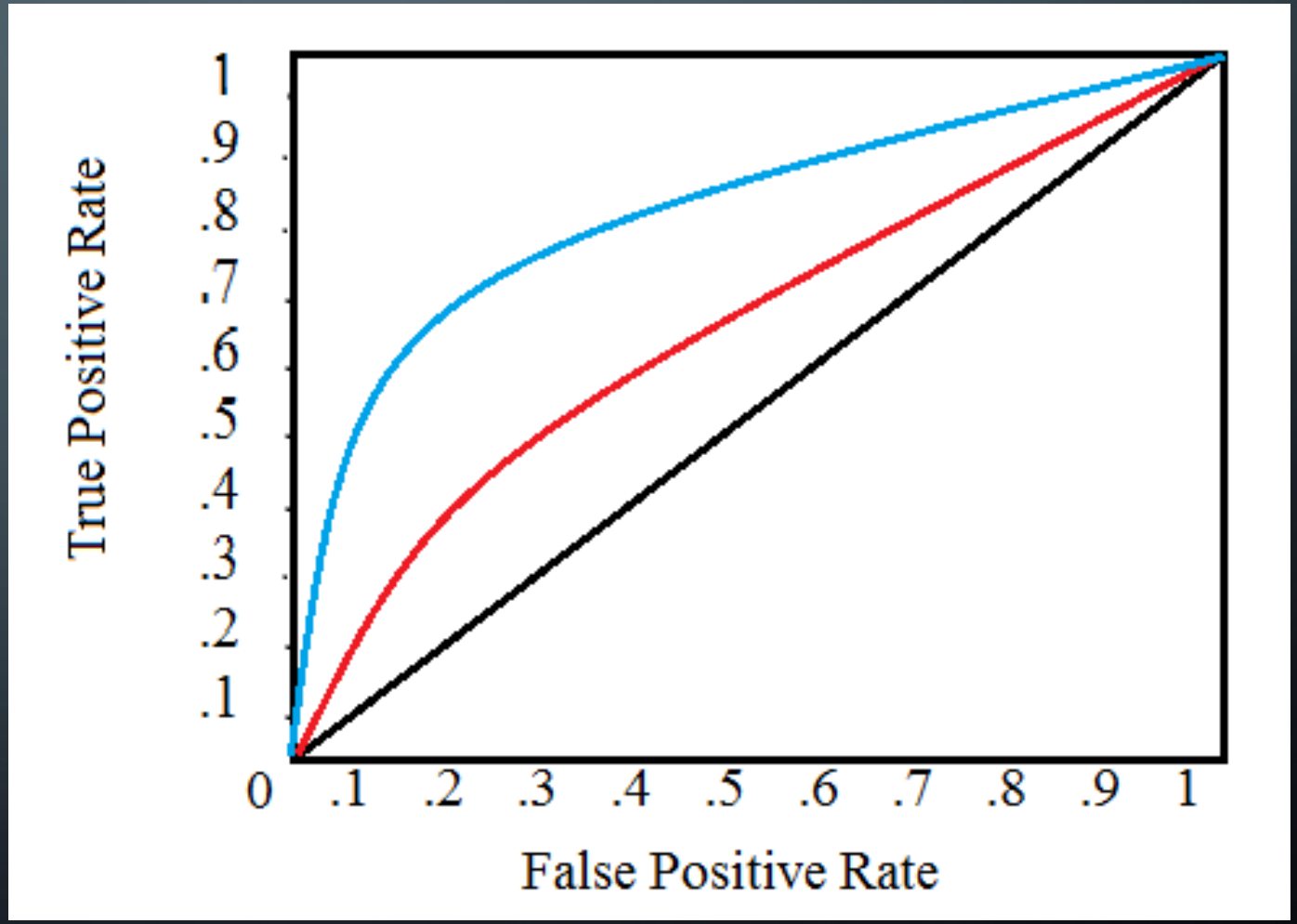
		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$F_1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

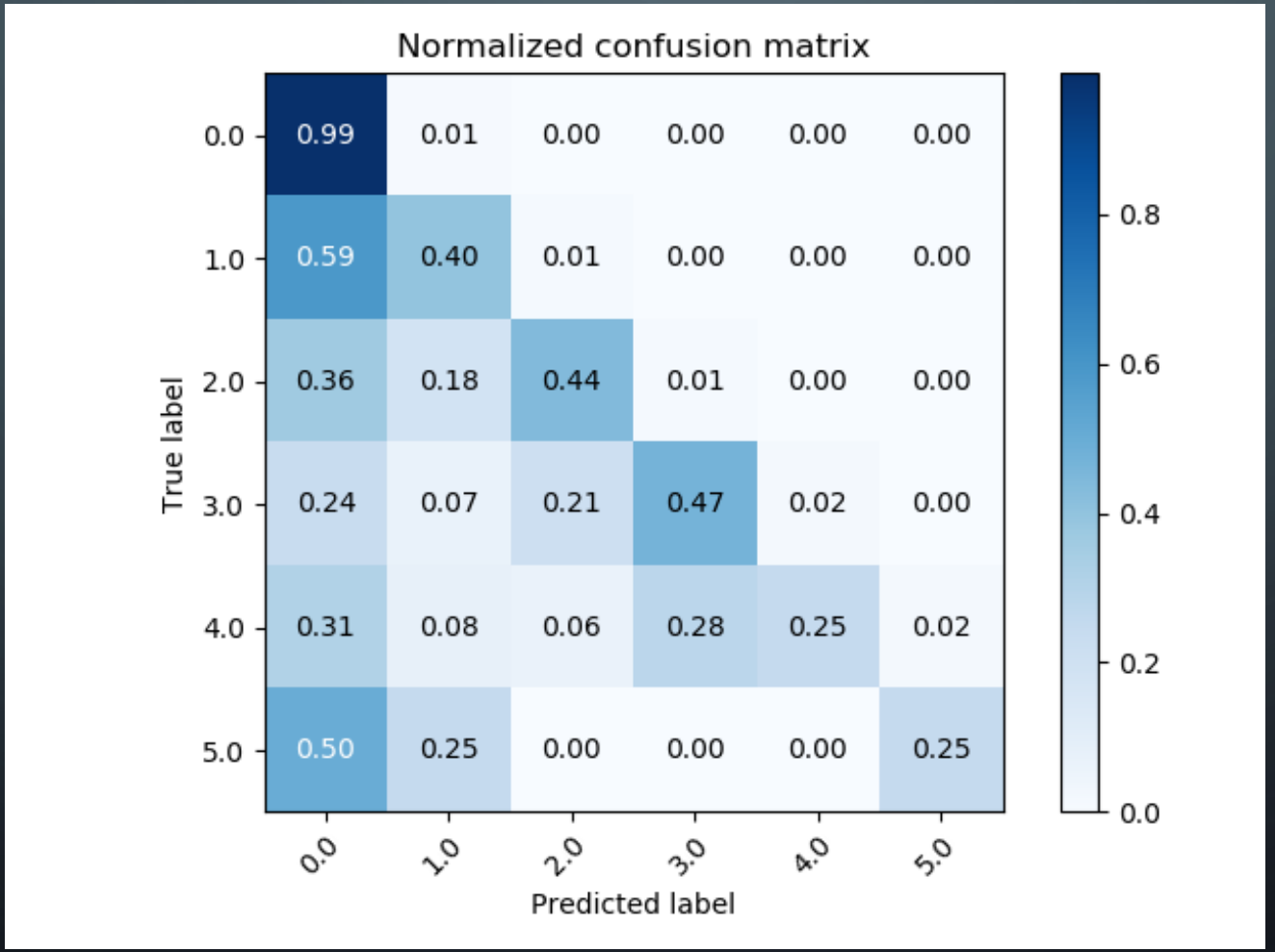
RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE

$$TPR = \frac{\#TP}{\#TP + \#FN}$$

$$FPR = \frac{\#FP}{\#FP + \#TN}$$



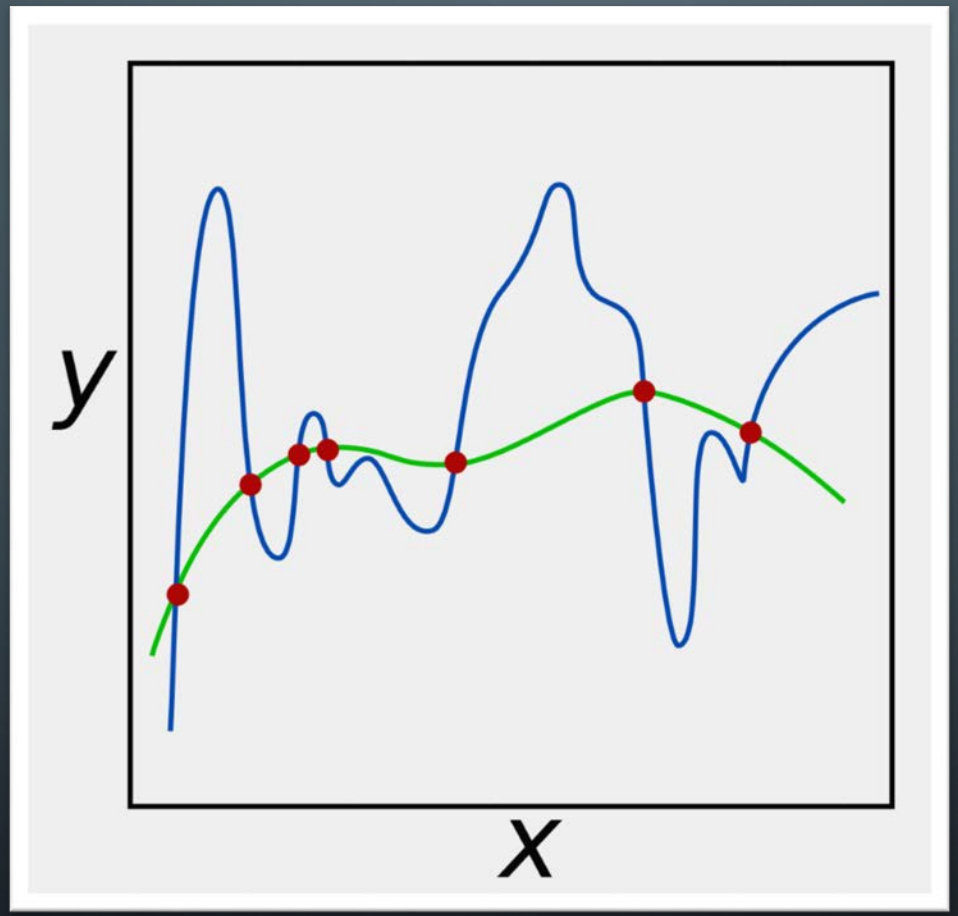
CONFUSION MATRIX



PERFORMANCE MEASURES

		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

REGULARIZATION



REGULARIZATIONS

- What causes overfitting:
  - Large weights (large derivatives)
  - Many features
  - Model trained too much
  - Powerful model
- Regularizations are techniques used to avoid overfitting.

REGULARIZATIONS

- Large weights (large derivatives)
  - L2 regularization
- Many features
  - L1 regularization
- Model trained too much
  - Early stopping
- Powerful model
  - dropout

L2 REGULARIZATION

- Large weights
  - So force model to chose small weights
- ridge regression

$$\hat{J} = J + \alpha \sum_{i=1}^{M} w_i^2$$
$$= J + \lambda \sum_{i=1}^{M} w_i^2$$
$$= J + \frac{1}{C} \sum_{i=1}^{M} w_i^2$$

class `sklearn.linear_model.` **`Ridge`**(*alpha=1.0, *, fit_intercept=True, normalize=False, copy_X=True, max_iter=None, tol=0.001, solver='auto', random_state=None*) ¶ [source]

L1 REGULARIZATION

- Many features
  - So force model to chose some features
- Lasso regression

$$\hat{J} = J + \alpha \sum_{i=1}^{M} |w_i|$$

class `sklearn.linear_model.` **`Lasso`**(*alpha=1.0, *, fit_intercept=True, normalize=False, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

EARLY STOPPING

- Stop model from training when validation loss is going upward