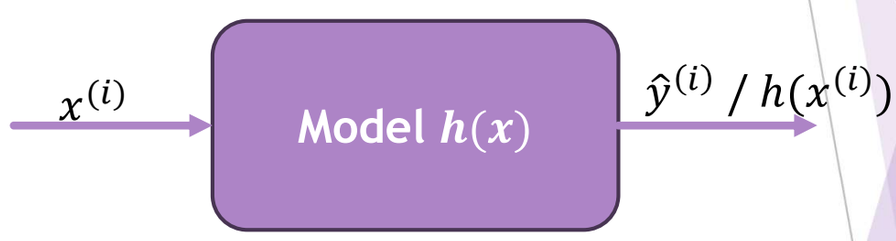
Evaluation and Model Selection

Maryam Abdolali

KNTU, Fall 2024

Select a Performance Measure

- Regression Problem:
  - Root Mean Square Error (RMSE)
    - $RMSE(X,h) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(h(x^{(i)}) - y^{(i)})^2}$
  - Mean Absolute Error (MAE)
    - $MAE(X,h) = \frac{1}{n}\sum_{i=1}^{n}\left|h(x^{(i)}) - y^{(i)}\right|$



- Classification Problem:
  - Accuracy (total #correct predictions/ total samples)

$$\text{acc} = \frac{1}{n}\sum_{x^{(i)}} I[h(x^{(i)}) = y^{(i)}]$$

  - Error rate (total incorrect predictions/ total samples)

$$\text{err} = \frac{1}{n}\sum_{x^{(i)}} I[h(x^{(i)}) \neq y^{(i)}]$$

Confusion Matrix

- **For a Binary classification problem**
  - Four possible outcome:
    - True Positive: TP = $\sum_{x^{(i)}} I[h(x^{(i)}) = y^{(i)} = \oplus]$
    - True Negative (TN): TN = $\sum_{x^{(i)}} I[h(x^{(i)}) = y^{(i)} = \ominus]$
    - False Positive (FP): FP = $\sum_{x^{(i)}} I[h(x^{(i)}) = \oplus, y^{(i)} = \ominus]$
    - False Negative (FN): FN = $\sum_{x^{(i)}} I[h(x^{(i)}) = \ominus, y^{(i)} = \oplus]$
  - Example in a diagnostic test to determine whether a person has a certain disease.
    - ✓ A false positive: person tests positive, but does not have the disease.
    - ✓ A false negative: person tests negative, suggesting they are healthy, when they actually do have the disease.

		Predicted	
		Positive (PP)	**Negative (PN)**
Actual	**Positive (P)**	TP	FN
	Negative (N)	FP	TN

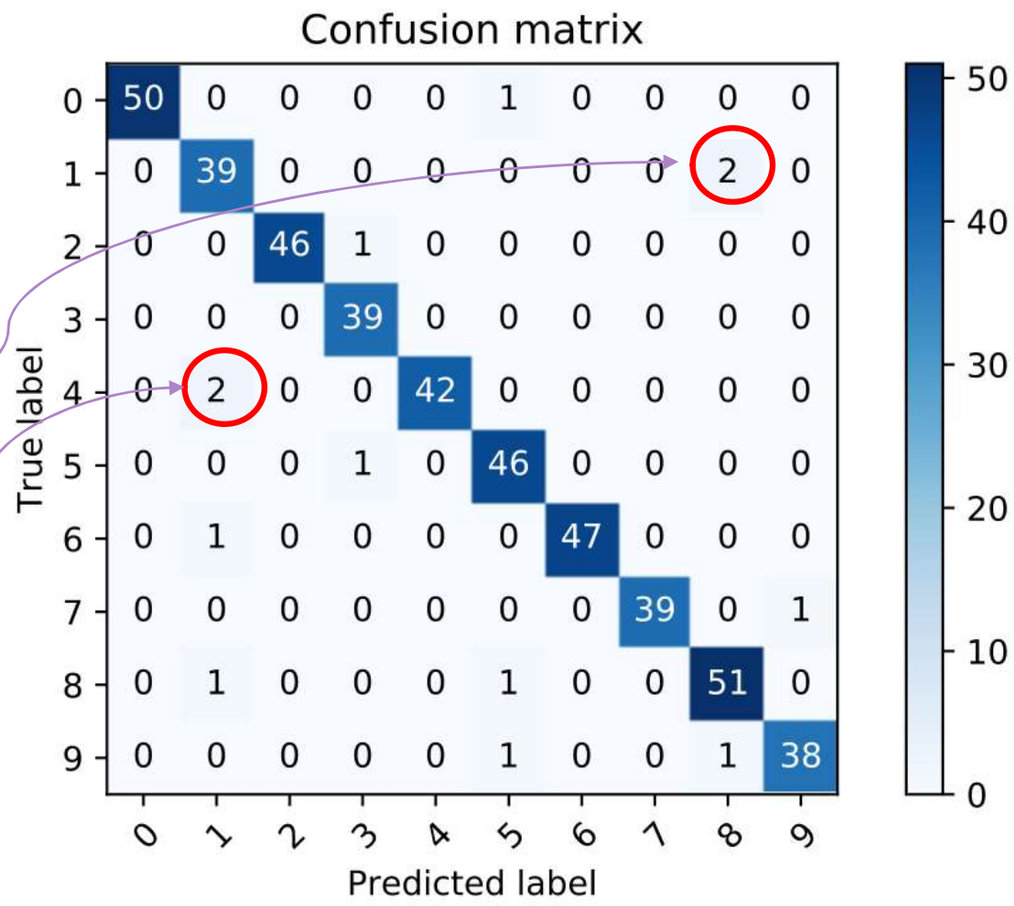
Example: Cancer vs Non-cancer classification

Individual #	1	2	3	4	5	6	7	8	9	10	11	12
Actual Classification	1	1	1	1	1	1	1	1	0	0	0	0
Predicted Classification	0	0	1	1	1	1	1	1	1	0	0	0
Result	FN	FN	TP	TP	TP	TP	TP	TP	FP	TN	TN	TN

		Predicted	
	Total: 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual	Cancer 8	6	2
	Non-cancer 4	1	3

Confusion matrix for multi-class classification

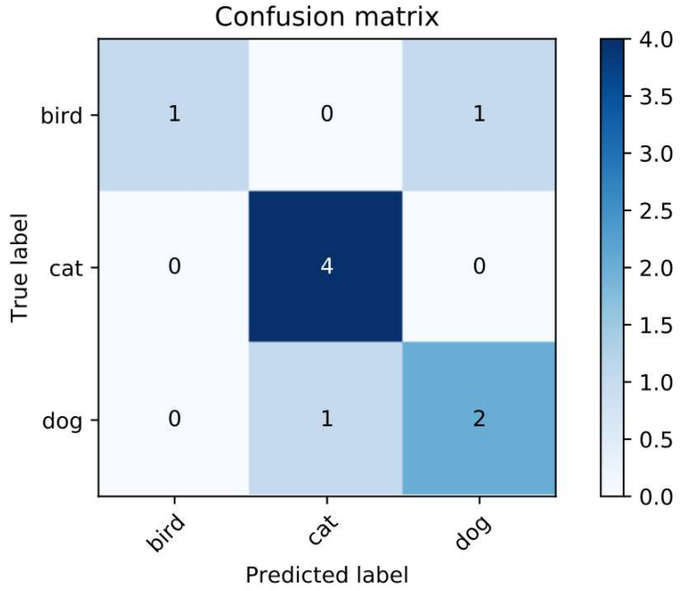
- Just like you generate a confusion matrix for a binary problem, you can generate one for a multi-class problem.
- As with the binary case, the rows represent the true labels, and the columns show the predicted labels.
- hand-written digits example
  - we can see that the model predicted 8 when the actual label was 1 two times,
  - The model predicted 1 when the label was 4 another couple of times
- However now we don't really have the notion of true negatives, false positives and so on.
- **The problem needs to be treated as a set of binary problems ("one-vs-all")**

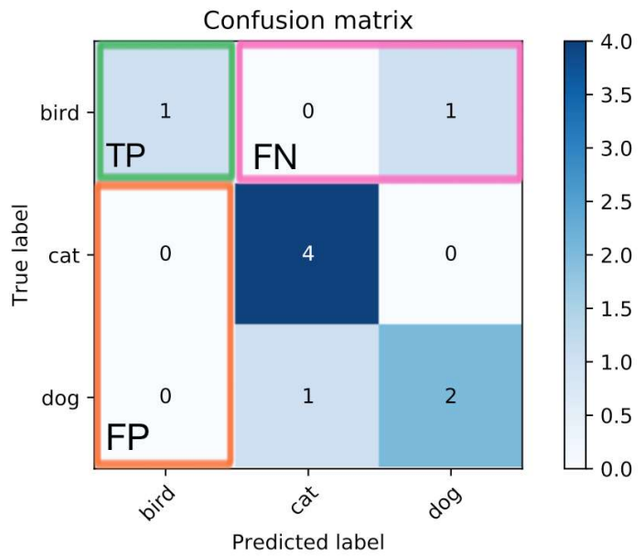


https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/

Example for multiclass classification

Label	Predicted
cat	cat
cat	cat
cat	cat
cat	cat
dog	dog
dog	dog
dog	cat
bird	dog
bird	bird





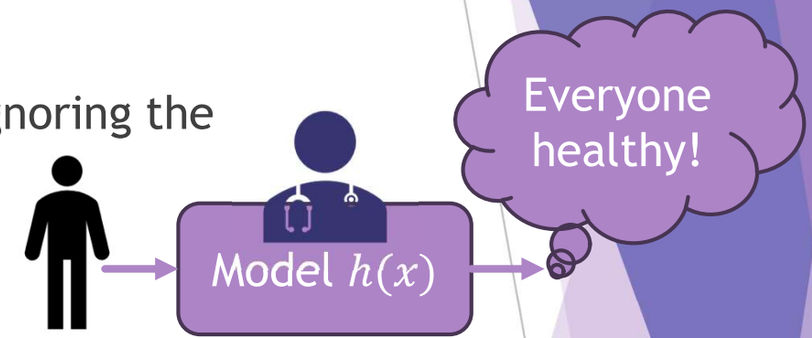
https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/

- **True Positives**: We only have one cell (highlighted green) where the true label was "bird" and the predicted label was "bird". The number in that cell will be the True Positives.
- **False Positives**: These are all those cases where "bird" was predicted, but the actual label was something else. These are all the cells in the same *column* as the true positives except the cell with the TP (highlighted orange). So, False Positives are the sum of the values in the orange area.
- **False Negatives**: These are all the times where the actual label was "bird" but the model predicted something else. These are all the cells in the same *row* as the true positives (highlighted pink) except the cell with TP. False Negatives is the sum of all those cells.

	TP	FP	FN
bird	1	0	1
cat	4	1	0
dog	2	1	1
TOTAL	7	2	2

Accuracy can be misleading…

- **Example Scenario: Total patients**: 1000
  - ✓ **Healthy (negative class)**: 950
  - ✓ **Cancer (positive class)**: 50
- Assume the classifier predicts that all patients are healthy, ignoring the cancer cases entirely.
- **Results:**
  - **True Negatives (healthy correctly identified)**: 950
  - **False Negatives (cancer misclassified as healthy)**: 50
  - **True Positives (cancer correctly identified)**: 0
  - **False Positives (healthy misclassified as cancer)**: 0
- **Calculating Accuracy:**
  - $\frac{TP+}{TP+TN+FN+FP} = \frac{950}{1000} = \%95$
- **Why is this Misleading in Cancer Prediction?**
  - **High accuracy** (95%) suggests the model is performing well, but in reality, it fails to identify **any cancer patients**. Cancer is the critical class in this context, and missing it could have serious consequences.



Performance measurement

- Precision (estimates $p(y = \oplus | \hat{y} = \oplus)$, TP/(TP+FP)) :

$$\frac{\sum_{x(i)} I[\hat{y}^{(i)} = y^{(i)} = \oplus]}{\sum_{x(i)} I[\hat{y}^{(i)} = \oplus]}$$

Precision is the ratio of correct positive predictions to the total positive predictions.
Out of all positives been predicted, how many are actually positive.

- Recall/ sensitivity (estimates $p(\hat{y} = \oplus | y = \oplus)$, TP/(TP+FN)) :

$$\frac{\sum_{x(i)} I[\hat{y}^{(i)} = y^{(i)} = \oplus]}{\sum_{x(i)} I[y^{(i)} = \oplus]}$$

Recall is a measure of how many positives your model is able to recall from the data.
Out of all positive records, how many records are predicted correctly.

Examples

Cancer Diagnosis

- Precision: Out of all the patients predicted to have cancer, how many actually have cancer?
  - Example: If a model predicts 100 patients have cancer, and 80 truly have cancer, the precision is 80%.
- Recall: Out of all the patients who actually have cancer, how many did the model correctly predict?
  - Example: If 200 patients actually have cancer, and the model identifies 150, the recall is 75%

Spam Email Detection

- Precision: Out of all the emails predicted to be spam, how many are actually spam?
  - Example: If the system flags 100 emails as spam, and 90 are indeed spam, the precision is 90%.
- Recall: Out of all the actual spam emails, how many did the system correctly flag?
  - Example: If there are 120 spam emails in total, and the system correctly flags 90, the recall is 75%.

Fraud Detection in Credit Card Transactions

- Precision: Out of all the transactions predicted to be fraudulent, how many are actually fraudulent?
  - Example: If the system flags 50 transactions as fraud, and 40 are indeed fraud, the precision is 80%.
- Recall: Out of all the actual fraudulent transactions, how many did the system correctly catch?
  - Example: If there are 60 fraudulent transactions in total, and the system catches 40, the recall is 67%.

Object Detection in Autonomous Cars

- Precision: Out of all the objects the car's system identifies as pedestrians, how many are actually pedestrians?
- Recall: Out of all the actual pedestrians, how many did the system correctly identify?

Defect Detection in Manufacturing

- Precision: Out of all the items predicted to have defects, how many actually have defects?
- Recall: Out of all the actual defective items, how many did the system catch?

Example for precision-recall calculation

- $True\ Positive\ (TP)\ =\ 50$
- $True\ Negative\ (TN)\ =\ 25$
- $False\ Positive\ (FP)\ =\ 15$
- $False\ Negative\ (FN)\ =\ 10$

- $Accuracy = \frac{TP+T}{Total} = \frac{50+2}{50+25+15+10} = 0.75$
- $Precision = \frac{TP}{TP+FP} = \frac{50}{50+1} = 0.77$
- $Recall = \frac{TP}{TP+F} = \frac{50}{50+1} = 0.83$

Unify Precision & Recall

- **F-score** conveys the balance between the precision and the recall in one number:
  - $F = \frac{2\ Precision \times Recall}{Precision + Recall}$

  - favors systems that achieve roughly equal precision and recall

	0.0	0.2	0.4	0.6	0.8	1.0
0.0	0.00	0.00	0.00	0.00	0.00	0.00
0.2	0.00	0.20	0.26	0.30	0.32	0.33
0.4	0.00	0.26	0.40	0.48	0.53	0.57
0.6	0.00	0.30	0.48	0.60	0.68	0.74
0.8	0.00	0.32	0.53	0.68	0.80	0.88
1.0	0.00	0.33	0.57	0.74	0.88	1.00

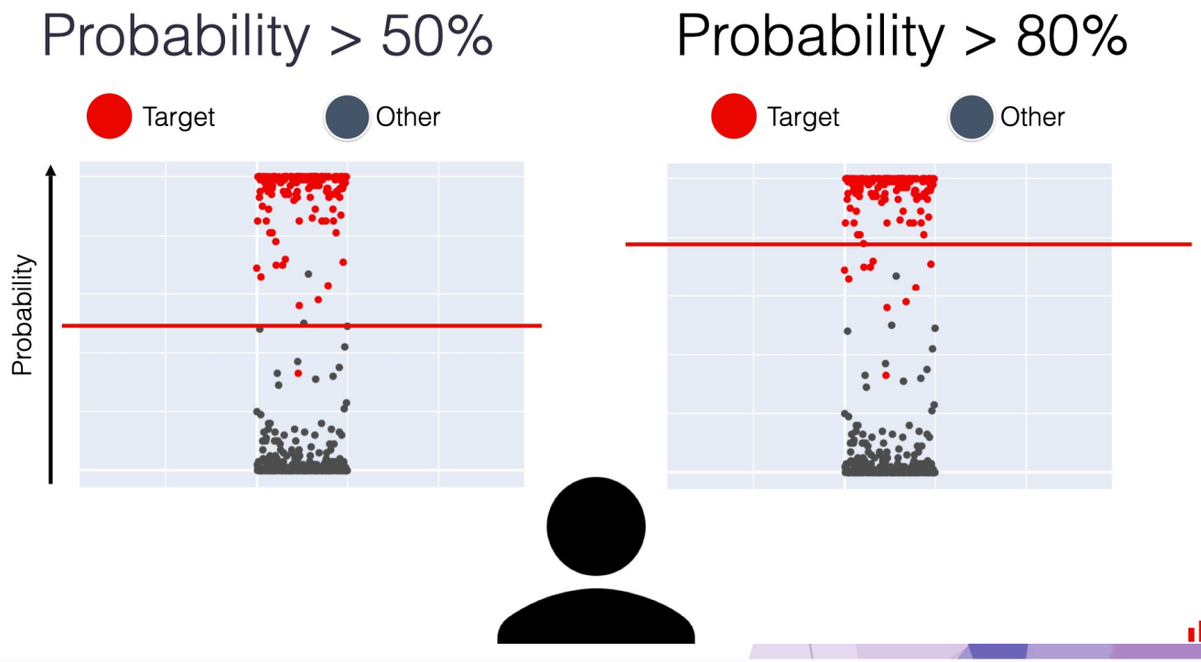
Table of f-measures when varying precision and recall values.

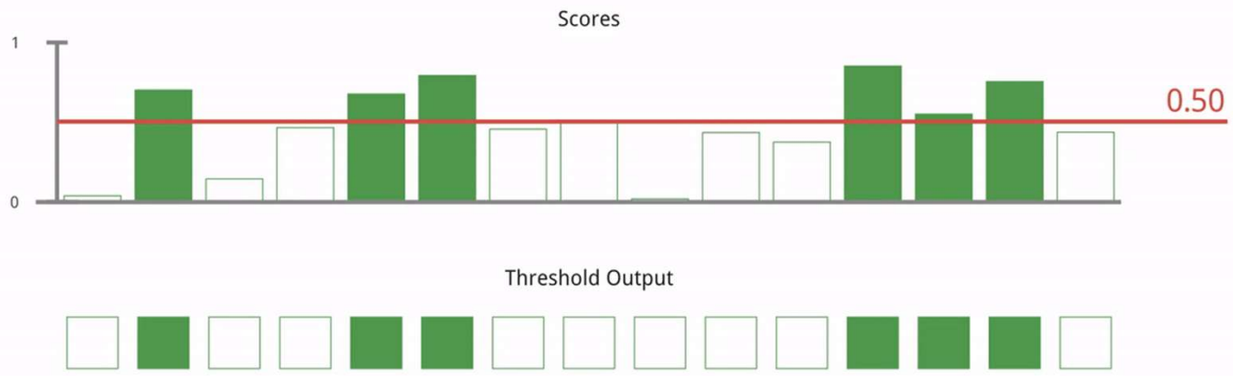
Precision-Recall trade-off

- Example 1- you trained a classifier to detect videos that are safe for kids
  - prefer a classifier that rejects many good videos (low recall)but keeps only safe ones (high precision), rather than a classifier that has a much higher recall but lets a few really bad videos show up in your product
- Example 2- suppose you train a classifier to detect shoplifters in surveillance images:
  - it is probably fine if your classifier only has 30% precision as long as it has 99% recall (sure, the security guards will get a few false alerts, but almost all shoplifters will get caught).
- Unfortunately, you can't have it both ways: increasing precision reduces recall, and vice versa. This is called the precision/recall trade-off.

-cont-

- Plotting precision-recall trade-off!
- Using Probability of Predictions
- A machine learning classification model:
  - ✓ directly predict the data point's actual class
  - ✓ predict its probability of belonging to different classes.
- The latter gives us more control over the result. We can determine *our own threshold* to interpret the result of the classifier.





https://www.evidentlyai.com/classification-metrics/explain-roc-curve

ROC curve

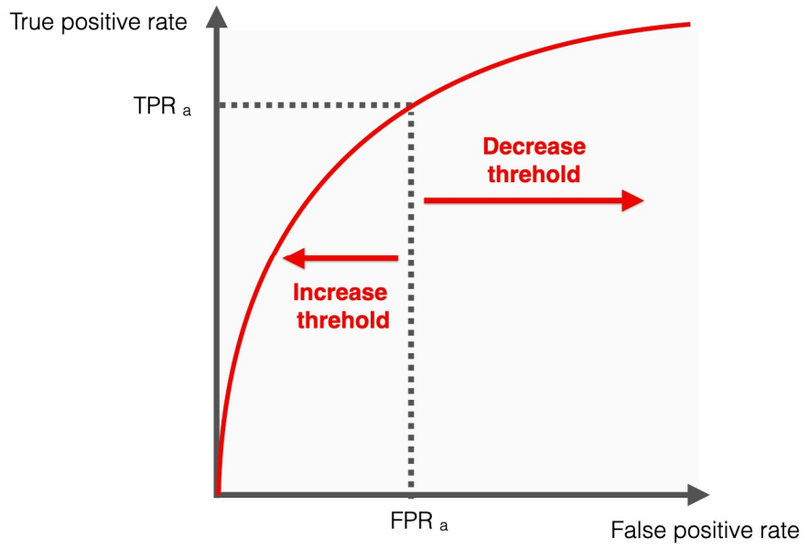
- Sensitivity/ Recall/ True Positive Rate (TPR):
  - The proportion of actual positives that are correctly classified

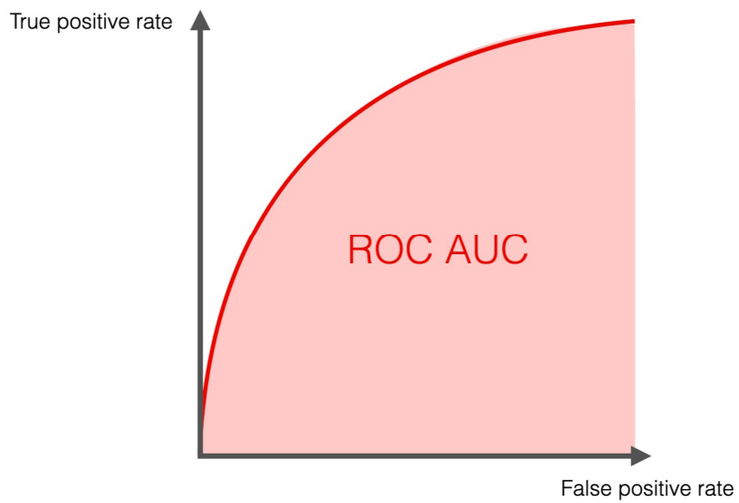
$$TPR = \frac{TP}{TP + FN}$$

- Specificity / False Positive Rate (FPR):
  - The proportion of actual negatives that are incorrectly classified as positive.

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

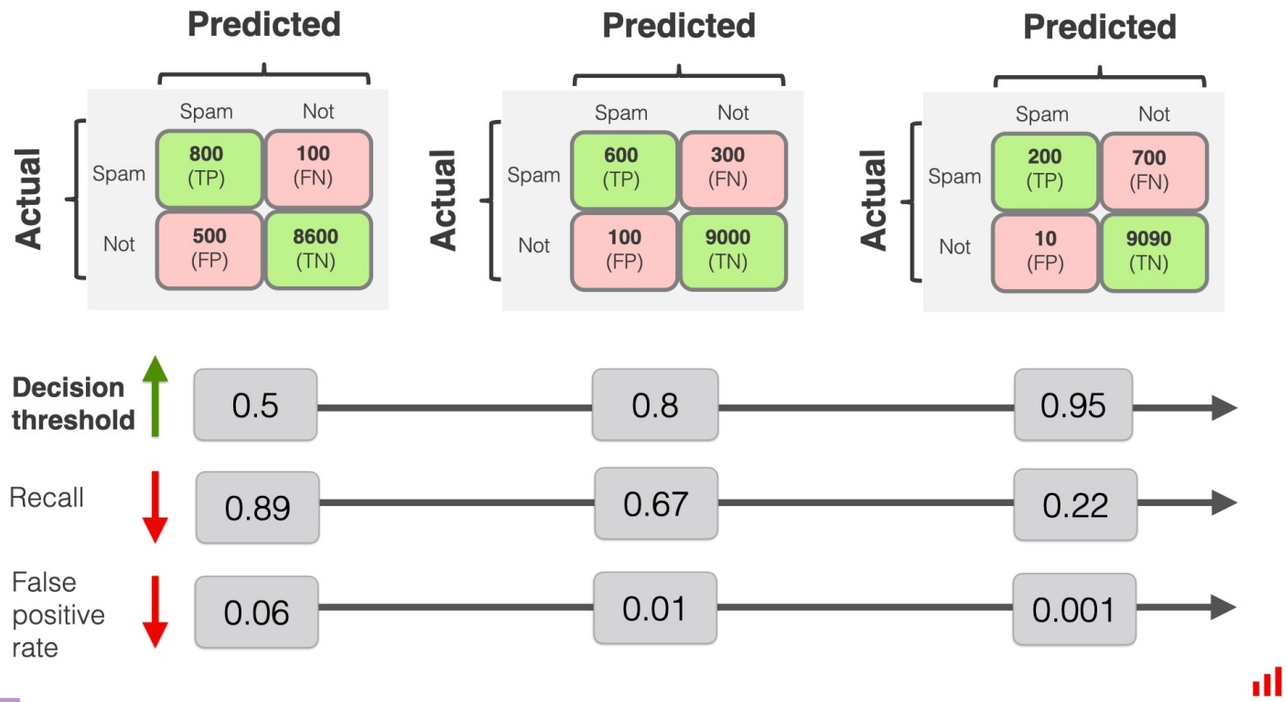
- To create the ROC curve, you need to plot the FPR values against TPR values at **different** decision thresholds.
- Given an ROC curve, you can compute the **area under the curve** (or **AUC**) metric, which also provides a meaningful single number for a system's performance.
- ***ROC curve*** can be used to **select a *threshold*** for a classifier that ***maximizes the true positives*** while ***minimizing the false positives.***

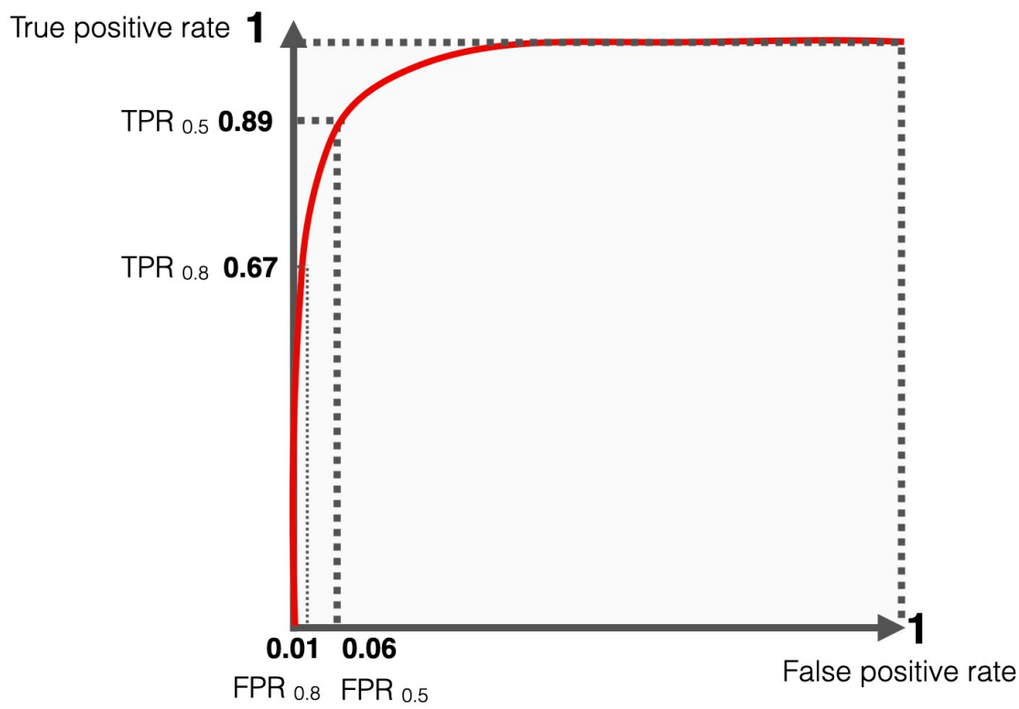




https://www.evidentlyai.com/classification-metrics/explain-roc-curve

Concept illustration

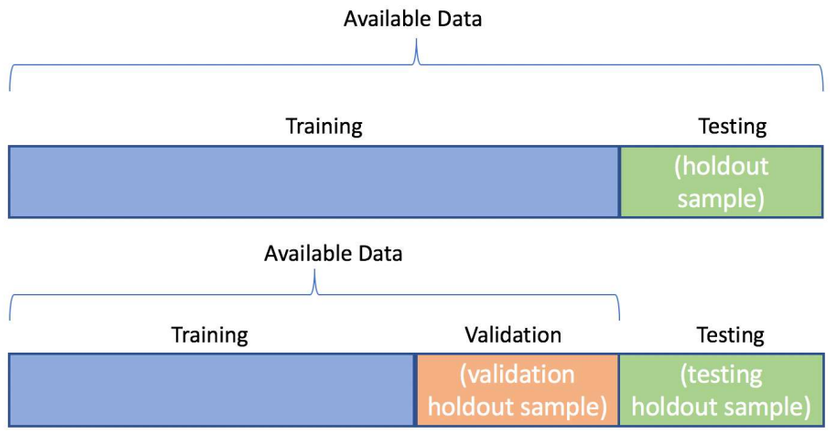




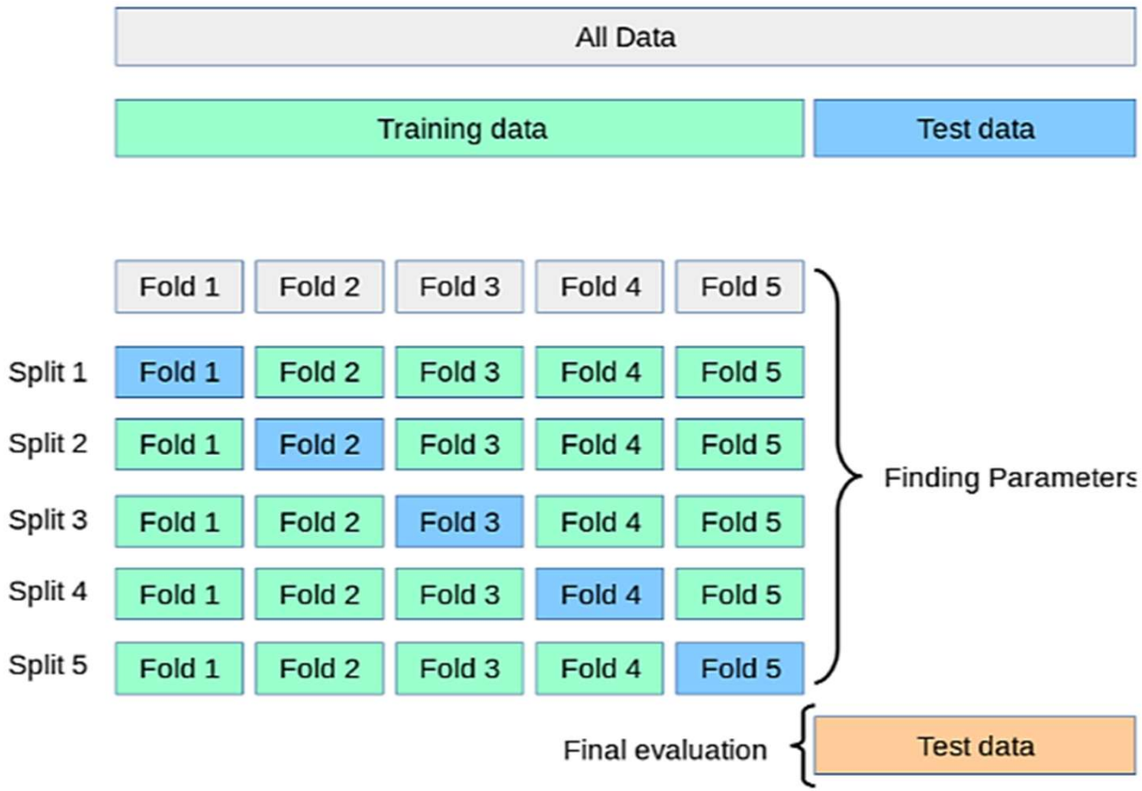
https://www.evidentlyai.com/classification-metrics/explain-roc-curve

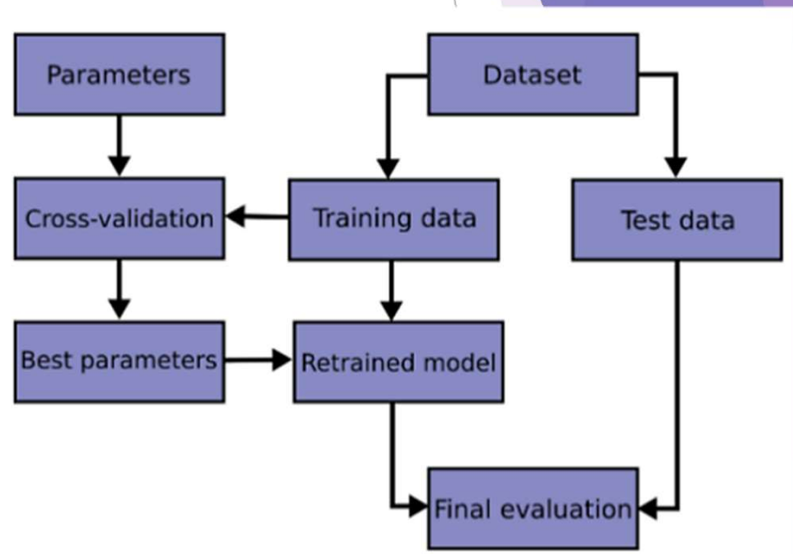
Hyperparameter selection (I) Validation

- How to tune hyperparameters?
- Why not using test data?
- The general approach:
  - ✓ 1. Split your data into ***70% training data, 10% validation data and 20% test data***.
  - ✓ 2. For each possible setting of your hyperparameters:
    - ✓ (a) Train a model using that setting of hyperparameters on the training data.
    - ✓ (b) Compute this model's error rate on the validation data.
  - ✓ 3. From the above collection of models, choose the one that achieved the *lowest error rate on validation data*.
  - ✓ 4. Evaluate that model on the test data to estimate future test performance.

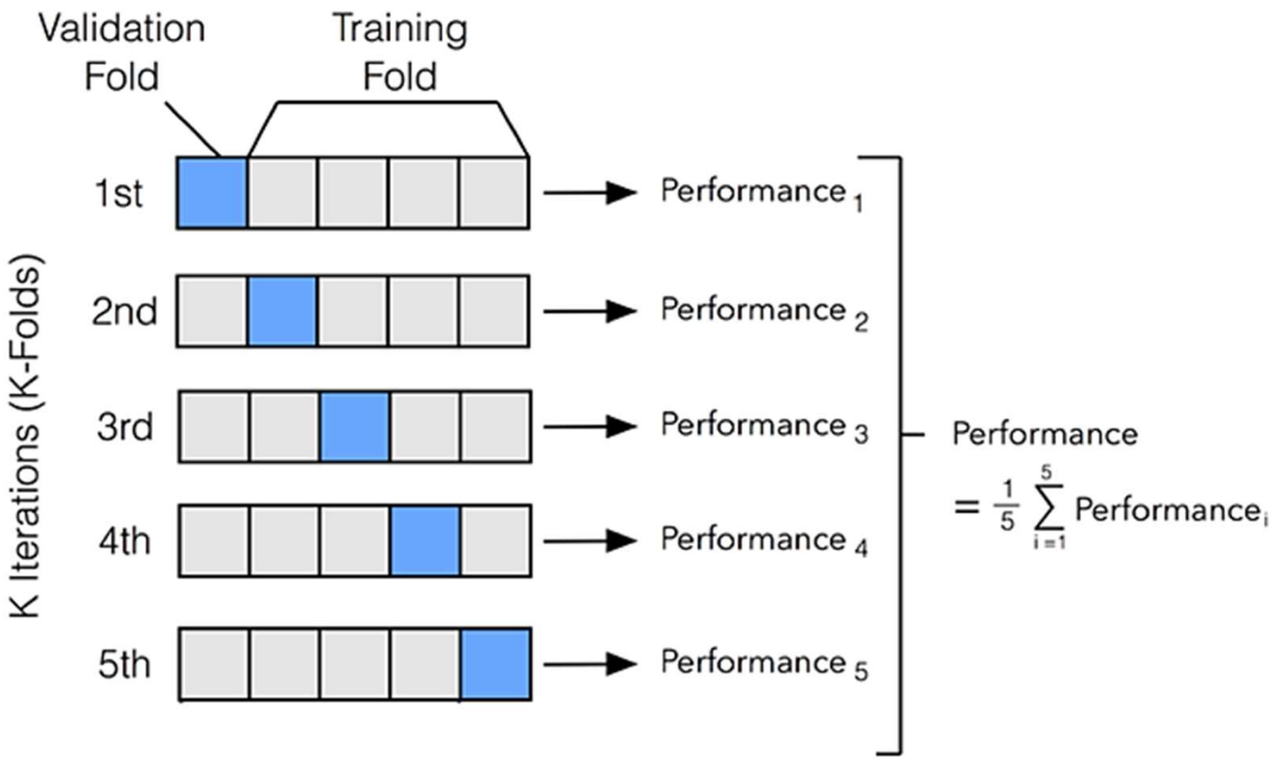


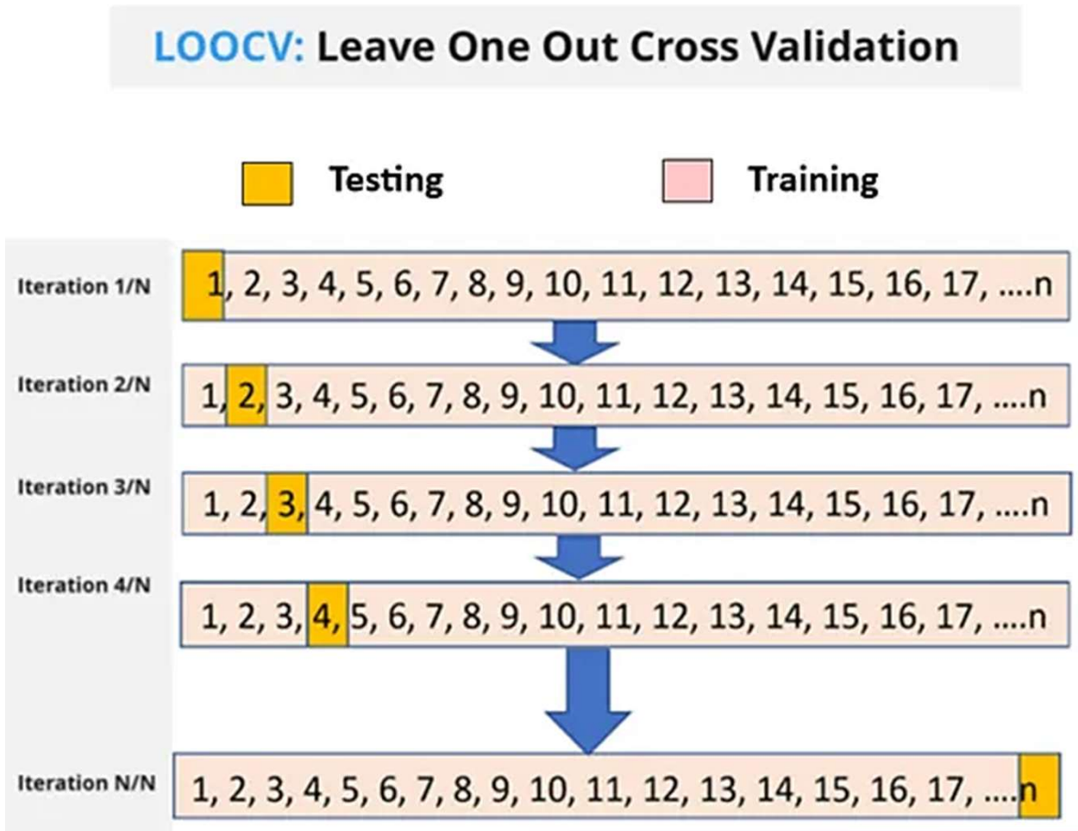
Model selection
(ii) cross-validation



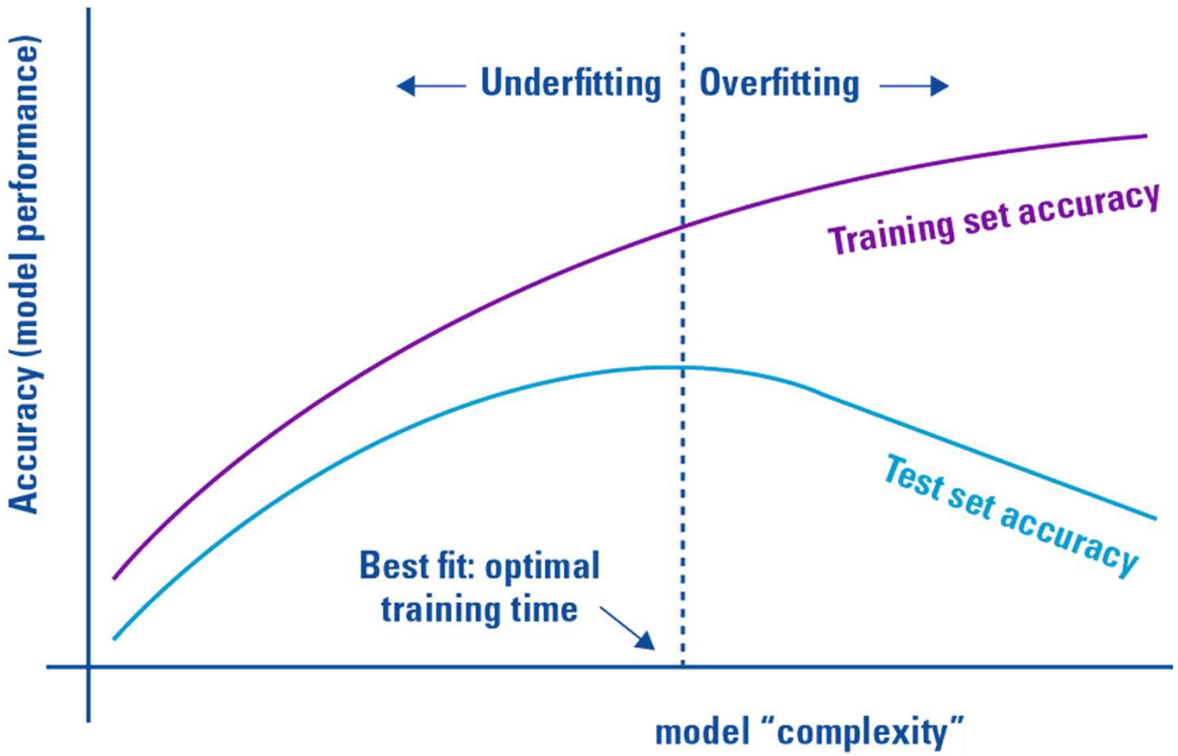


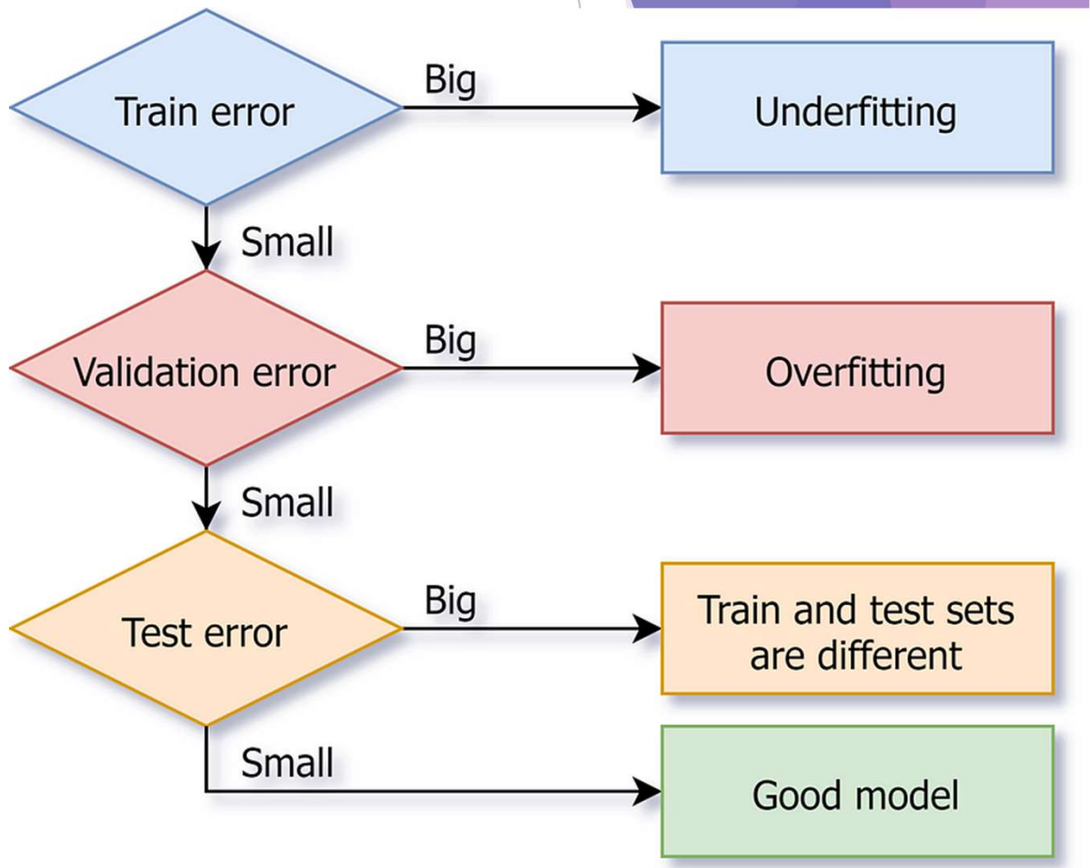
K-Fold cross-validation
Leave-one-out validation





Underfit vs Overfit





Example for 3fold cross-validation

- We'll use a dataset with 12 samples and a simple binary classification task. The dataset looks like this:

Sample	Feature	Class (True Label)
S1	2	0
S2	3	0
S3	4	1
S4	5	1
S5	6	0
S6	7	0
S7	8	1
S8	9	1
S9	10	0
S10	11	0
S11	12	1
S12	13	1

solution

- We'll split the 12 samples into 3 folds (F1, F2, F3), each containing 4 samples:
  - **Fold 1 (F1)**: S1, S2, S3, S4
  - **Fold 2 (F2)**: S5, S6, S7, S8
  - **Fold 3 (F3)**: S9, S10, S11, S12
- **Step 2: Perform Cross-Validation**
  - **Iteration 1 (Fold 1 as the Validation Set):**
  - **Training Set**: F2 + F3 (Samples: S5, S6, S7, S8, S9, S10, S11, S12)
  - **Validation Set**: F1 (Samples: S1, S2, S3, S4)
  - Assume the model's predictions for the validation set (F1) are as follows:
  - Accuracy (F1)=0.75

Sample	True Label	Prediction
S1	0	0
S2	0	1
S3	1	1
S4	1	1

-cont-

- **Iteration 2 (Fold 2 as the Validation Set):**
  - **Training Set**: F1 + F3 (Samples: S1, S2, S3, S4, S9, S10, S11, S12)
  - **Validation Set**: F2 (Samples: S5, S6, S7, S8)
  - Assume the model's predictions for the validation set (F2) are:
  - The accuracy for Fold 2 is:
  - Accuracy (F2)=0.75

Sample	True Label	Prediction
S5	0	0
S6	0	0
S7	1	1
S8	1	0

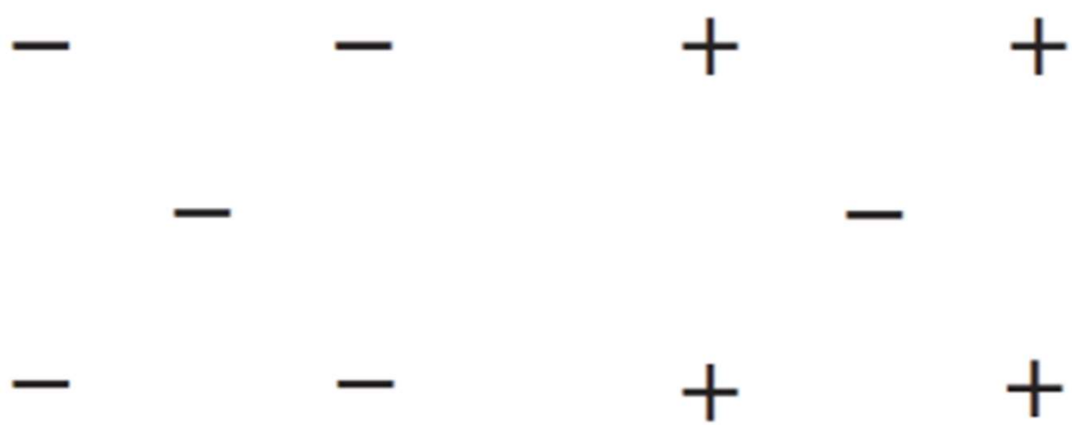
- **Iteration 3 (Fold 3 as the Validation Set):**
  - **Training Set**: F1 + F2 (Samples: S1, S2, S3, S4, S5, S6, S7, S8)
  - **Validation Set**: F3 (Samples: S9, S10, S11, S12)
  - Assume the model's predictions for the validation set (F3) are:
  - The accuracy for Fold 3 is:
  - Accuracy (F3)=1

Sample	True Label	Prediction
S9	0	0
S10	0	0
S11	1	1
S12	1	1

- **Step 3: Average the Results**
  - Average Accuracy=$\frac{0.75 + 0.75 + 1.00}{3} = 0.833$

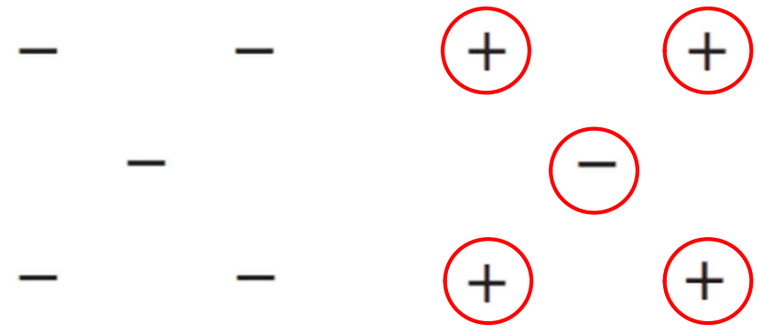
Example

- What is the leave-one-out cross-validation error on this dataset for 1NN and 3NN?



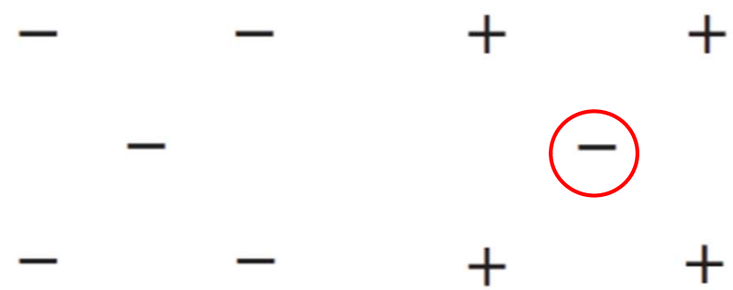
Solution

1NN



# of errors= 0+0+0+0+0+1+1+1+1+1=5
Error = 5/10

3NN



# of errors = 0+0+0+0+0+0+0+0+0+1=1
Error = 1/10