



# CAP 4630 – Evaluation Metrics

**Instructor:** Aakash Kumar

University of Central Florida

# Flow of Batch Machine Learning

## ➤ Given:

- Labeled training data  $(X, Y)$ , where  $X$  represents input features and  $Y$  represents corresponding labels.
- Assumes each data point  $X_i$  is drawn from a distribution  $D(X)$ , with its label  $y_i$  generated by a target function  $f_{\text{target}}(x_i)$ .

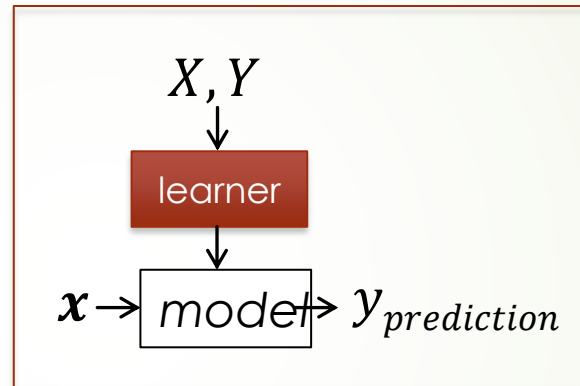
## ➤ Train the model:

- The classifier learns patterns in the labeled training data  $(X, Y)$  to build a model that captures the relationship between input features and their corresponding labels.
- $\text{Model} \leftarrow \text{classifier.train}(X, Y)$

## ➤ Apply the model to new data:

- Given a new, unseen instance  $x$ , drawn from the same distribution  $D(X)$ , the trained model generates a prediction.
- $y_{\text{prediction}} \leftarrow \text{model.predict}(x)$

# Flow of Batch Machine Learning



## ► Key questions:

- How do we evaluate the model's performance?
- What metrics should we use? (e.g., accuracy, precision, recall, F1-score)
- How does this model compare to others? (e.g., cross-validation, comparison to baseline models)

# Metrics

- ▶ We train our model on the training data:  $\text{Train} = \{x_i, y_i\}_{i=1,m}$ .
- ▶ We evaluate performance on test data.
- ▶ We often set aside part of the training data as a development set for hyperparameter tuning and model refinement.
- ▶ For binary classification tasks, we commonly use accuracy as a performance metric:

$$\text{accuracy} = \frac{\text{\#correct predictions}}{\text{\#test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\text{\#incorrect predictions}}{\text{\#test instances}}$$

# Example of Computing Accuracy

- Example 1: Balanced Dataset

- Suppose we have 10 test instances, and the model makes 8 correct predictions.

$$\text{accuracy} = \frac{8}{10} = 0.80 \text{ or } 80\%$$

- In this case, accuracy gives a good sense of model performance since the dataset is balanced, and 80% of predictions are correct.

- Example 2: Another Balanced Dataset

- In another test, we have 20 test instances, and the model makes 18 correct predictions.

$$\text{accuracy} = \frac{18}{20} = 0.90 \text{ or } 90\%$$

- Once again, accuracy provides a reliable measure of performance in a balanced dataset.

# Why Accuracy is Not Always a Good Metric

## ➤ Example 3: Imbalanced Dataset

- Consider a dataset with 100 test instances where 95 instances belong to class A and 5 to class B.
- A model predicts all instances as class A and none from class B.

$$\text{accuracy} = \frac{95}{100} = 0.95 \text{ or } 95\%$$

- Issue: The model achieves high accuracy but completely fails to predict class B. This is misleading in cases where we care about detecting the minority class.

# Why Accuracy is Not Always a Good Metric

## ➤ Example 4: Severe Class Imbalance

- In an extreme case, the dataset has 1,000 instances: 990 from class A and 10 from class B.
- The model predicts all as class A.

$$\text{accuracy} = \frac{990}{1000} = 0.99 \text{ or } 99\%$$

- Issue: The model ignores class B entirely, but still reports a 99% accuracy. In such cases, accuracy is not a useful metric for evaluating performance.



# Alternative Metrics

- If the Binary classification problem is biased
  - In many problems most examples are negative
- Or, in multiclass classification
  - The distribution over labels is often non-uniform
- Simple accuracy is not a useful metric.
  - Often, we resort to task specific metrics
- However, one important example that is being used often involves **Recall** and **Precision**



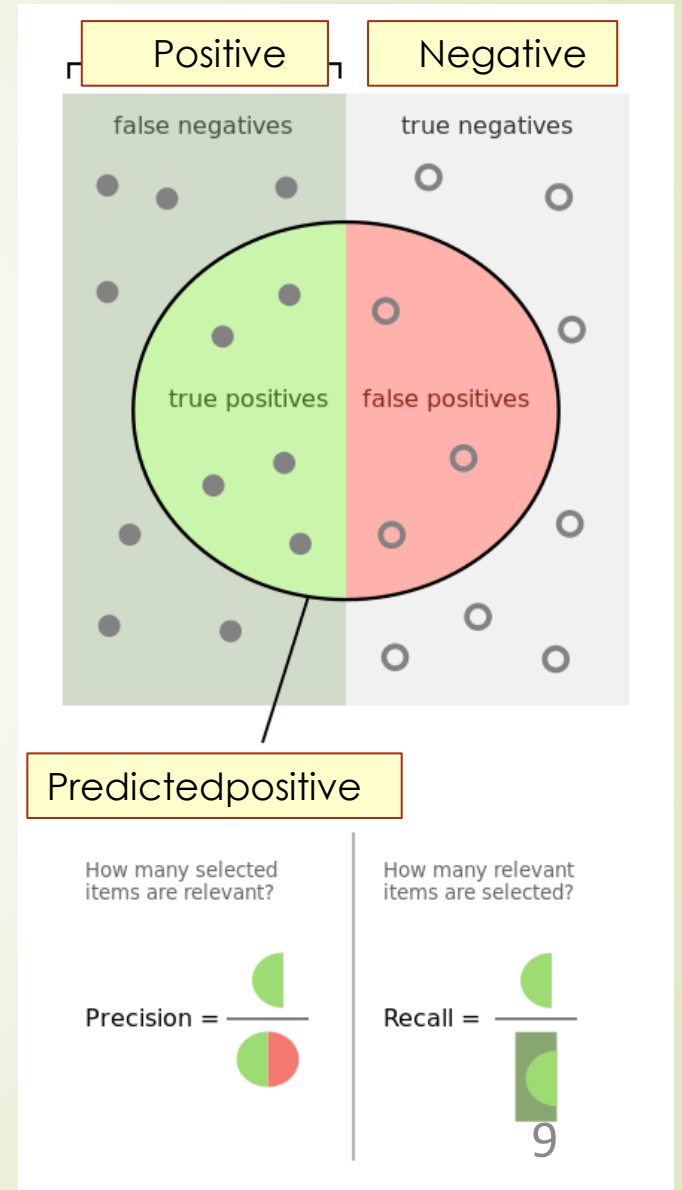
# Alternative Metrics

- **Recall:** Measures the ability to identify **all relevant instances** (sensitivity).

$$\text{Recall} = \frac{\# \text{True Positives}}{\# \text{All Positives}}$$

- **Precision:** Measures how many of the predicted positives are **actually positive**.

$$\text{Precision} = \frac{\# \text{True Positives}}{\# \text{Predicted Positives}}$$



# Example

- 100 examples, 5% are positive.
- Just say NO: your accuracy is 95%
  - Recall = precision = 0
- Predict 4+, 96-; 2 of the +s are indeed positive
  - Predicted Positive: 4 (2 correct, 2 incorrect)
  - Predicted Negative: 96

$$\text{Recall} = \frac{\# \text{True Positives}}{\# \text{All Positives}}$$

- Recall: 2/5 -----> 40%
- Precision: 2/4 -----> 50%

Confusion Matrix:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (2)	False Negative (3)
Actual Negative	False Positive (2)	True Negative (94)

Accuracy Calculation:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}}$$
$$\text{Accuracy} = \frac{2 + 94}{100} = \frac{96}{100} = 96\%$$

$$\text{Precision} = \frac{\# \text{True Positives}}{\# \text{Predicted Positives}}$$

# Confusion Metrics

- Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

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

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

# Example 1 - Balanced Dataset (Accuracy Works Well)

## Scenario:

- Dataset: 100 total examples
- 50 positives and 50 negatives
- Model predicts 45 positives correctly and 45 negatives correctly

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (45)	False Negative (5)
Actual Negative	False Positive (5)	True Negative (45)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} = \frac{45 + 45}{100} = 90\%$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{45}{45 + 5} = 90\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{45}{45 + 5} = 90\%$$

## Example 2 - Imbalanced Dataset (Accuracy Fails)

### Scenario:

- Dataset: 100 total examples
- 5 positives and 95 negatives
- Model predicts all examples as negative (ignores positives)

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (0)	False Negative (5)
Actual Negative	False Positive (0)	True Negative (95)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} = \frac{0 + 95}{100} = 95\%$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{0}{0 + 0} = N/A \text{ (no positives predicted)}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{0}{0 + 5} = 0\%$$

# Example 3 - Imbalanced Dataset (Precision & Recall Work)

## Scenario:

- Dataset: 100 total examples
- 5 positives and 95 negatives
- Model predicts 4 positives, 3 are correct, and 96 negatives

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (3)	False Negative (2)
Actual Negative	False Positive (1)	True Negative (95)

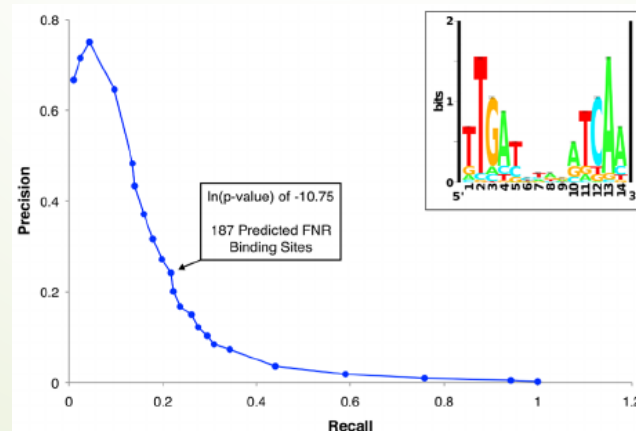
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Instances}} = \frac{3 + 95}{100} = 98\%$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{3}{3 + 1} = 75\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{3 + 2} = 60\%$$

# Relevant Metrics

- Combining Precision and Recall:
  - It makes sense to consider Recall and Precision together, as they provide complementary information about a model's performance.
  - These metrics can be combined into a single performance measure for a more holistic view.
- Recall-Precision Curve:
  - Definition: A plot that shows the trade-off between precision and recall at different thresholds.
  - Interpretation: As recall increases, precision typically decreases. The curve helps evaluate how well the model balances precision and recall.
  - Usage: Particularly useful in imbalanced datasets where accuracy might be misleading.





# How to Combine Precision and Recall

## Generalized $F_\beta$ -Measure

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}}$$

- $F_\beta$  allows you to adjust the balance between precision and recall based on the problem requirements:

- When  $\beta > 1$ , more emphasis is placed on Recall.

- When  $\beta < 1$ , more emphasis is placed on Precision.

## ■ F1-Score (F-Measure):

- Definition: The harmonic mean of precision and recall, combining them into a single metric. F1 gives more weight to low values, so if either precision or recall is low, the F1 score will also be low.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- F1 Score: The most commonly used F-Measure, balances precision and recall equally.

# Comparing Classifiers

- ▶ Can we use training accuracy to choose between them?
  - ▶ No!
  - ▶ Why? Training accuracy may be misleading because it reflects the model's performance on the data it has already seen, often leading to overfitting.
- ▶ What about accuracy on test data?
  - ▶ Yes, but...
  - ▶ Accuracy alone isn't enough. We need to consider other metrics (e.g., precision, recall, F1-score) to get a more complete picture of performance.
  - ▶ Statistical significance: It's crucial to perform statistical tests (e.g., cross-validation) to ensure the difference in performance is not due to random chance.
  - ▶ Look at multiple metrics: Evaluate classifiers using metrics that align with the task's goals and the data's characteristics, such as handling class imbalances.

# K-fold Cross Validation

- Instead of a single test-training split:

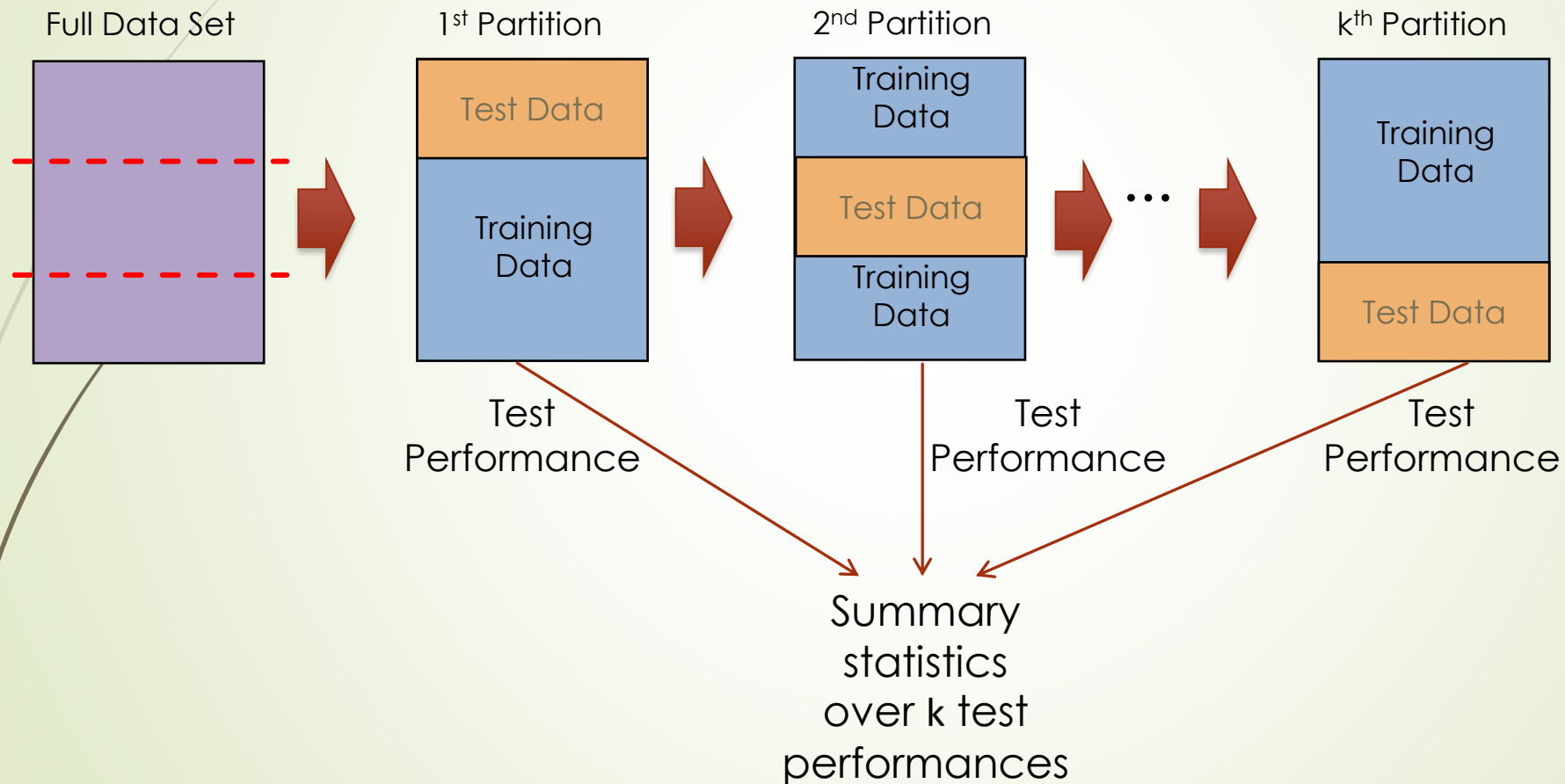


- Split data into K equal-sized parts



- Train and test K different classifiers
- Report average accuracy and standard deviation of the accuracy

# Example k-Fold CV

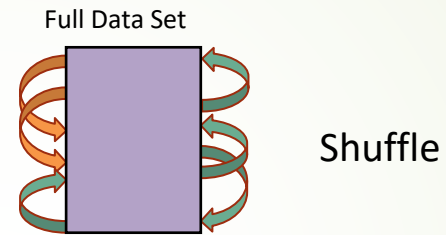


# More on Cross-Validation

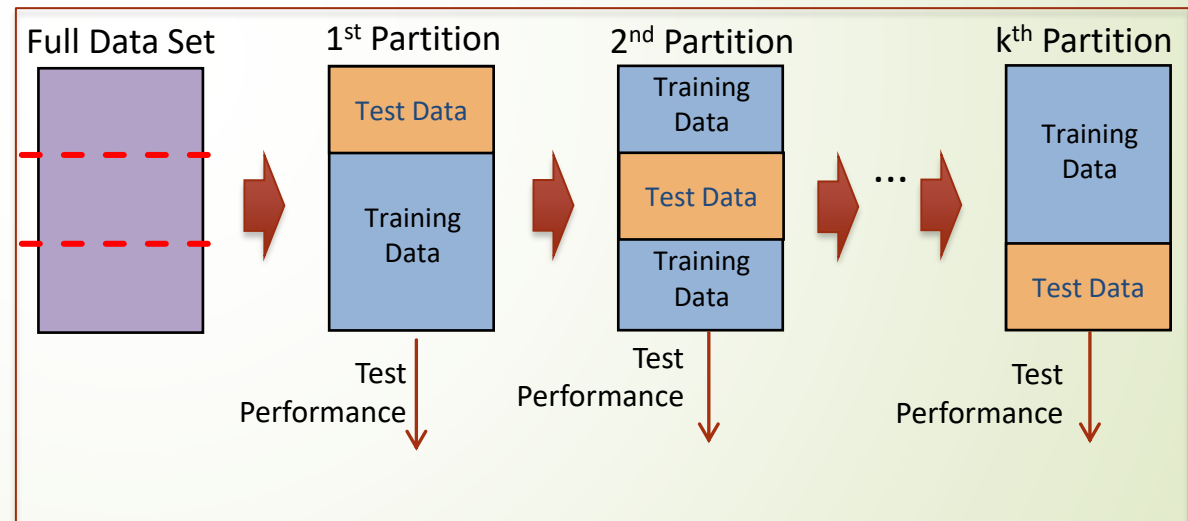
- Purpose: Cross-validation provides an approximate estimate of how well the classifier will perform on unseen data.
  - As  $K$  approaches  $N$ : The model becomes more accurate since it trains on more data.
  - Trade-off: As  $K$  increases, computational cost increases, making cross-validation more expensive.
  - Choosing  $K < N$ : A common compromise is  $K = 5$  or  $K = 10$ , which balances accuracy and efficiency.
  - Leave-One-Out Cross-Validation (LOO-CV): When  $K = N$ , each instance is used once as the validation set. This can be useful for small datasets but is computationally intensive.

# Multiple Trials of k-Fold CV

- Loops for  $t$  trails
  - Randomize Data Set



- Perform k-fold CV



- Compute statistics over  $t \times k$  test performances

# Evaluation: significance tests

- You have two different classifiers, A and B
- You train and test them on the same data set using N-fold cross-validation
- For the n-th fold:  
     $\text{accuracy}(A, n), \text{accuracy}(B, n)$   
     $p_n = \text{accuracy}(A, n) - \text{accuracy}(B, n)$
- Is the difference between A and B's accuracies significant?



# Evaluation: Significance Tests

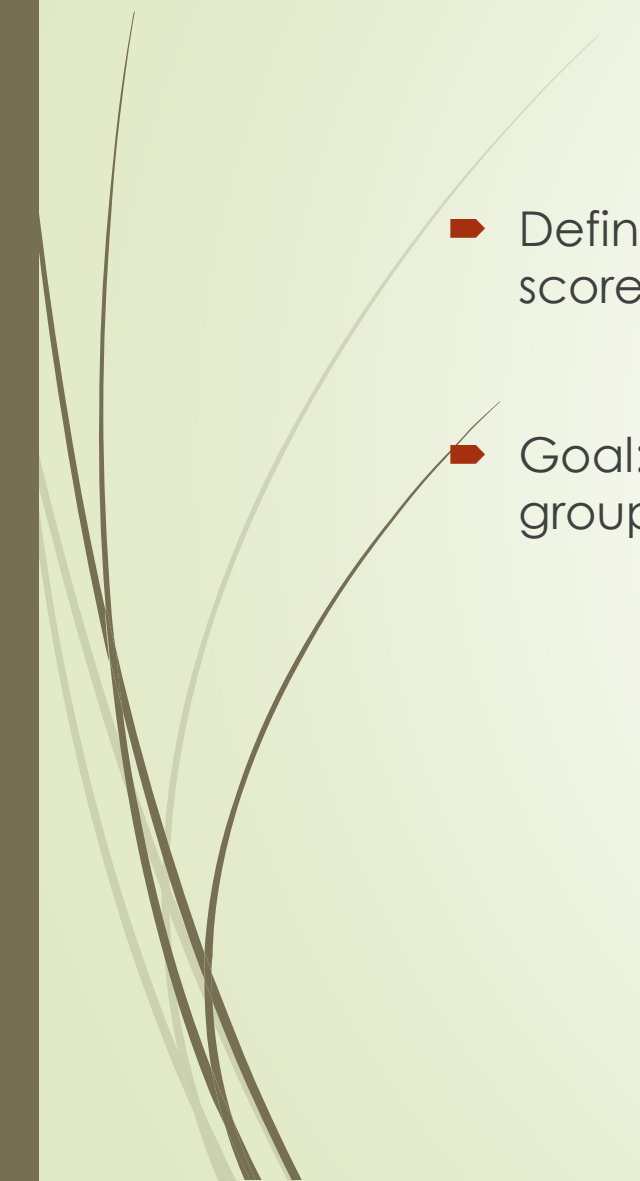
- Goal: Determine whether the observed difference in accuracy between two classifiers (A and B) is statistically significant.
- Process:
  - Train and Test: Use N-fold cross-validation to train and test both classifiers on the same dataset.
- Calculate Performance: For each fold  $n$ :
- Record accuracies:  $\text{accuracy}(A, n)$  and  $\text{accuracy}(B, n)$ .

Calculate the difference:  $p_n = \text{accuracy}(A, n) - \text{accuracy}(B, n)$ .

- Key Question:
  - Is the observed difference in accuracy between classifiers A and B statistically significant, or could it have occurred by chance?



# Introduction to Paired t-Test

- Definition: The paired t-test compares two related groups, like before-and-after scores or results from two different methods applied to the same dataset.
  - Goal: It helps to find out if there's a significant difference between the two groups.
- 

# Hypothesis Testing for the Paired t-Test

- Define Hypothesis H: This is what you're trying to prove (e.g., "Classifier A performs differently from Classifier B").
- Define Null Hypothesis  $H_0$  : This is the opposite of what you're trying to prove (e.g., "There is no difference between Classifier A and B").
- Goal: We need to determine if we can reject the null hypothesis (i.e., show that there is a difference between the two classifiers).



# Example: Comparing Two Classifiers

Score from Classifier A	Score from Classifier B
80	85
75	80
90	88
...	...

# Steps to Perform a Paired t-Test

- Step 1: Calculate the difference between each pair of scores (e.g., Classifier A - Classifier B for each subject).
- Step 2: Calculate the mean of these differences (this tells us the average difference).
- Step 3: Find the standard deviation of these differences (how spread out the differences are).
- Step 4: Use the formula to calculate the t-statistic:

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$$

- $\bar{d}$  = mean difference
- $s_d$  = standard deviation of the differences
- $n$  = number of pairs

- Step 5: Compare the t-statistic to a critical value (from a t-distribution table). If the t-statistic is larger than the critical value, the difference is significant.

# Paired t-test example

Subject #	Score 1	Score 2
1	3	20
2	3	13
3	3	13
4	12	20
5	15	29
6	16	32
7	17	23
8	19	20
9	23	25
10	24	15
11	32	30

# Paired t-test

Subject #	Score 1	Score 2	X-Y
1	3	20	-17
2	3	13	-10
3	3	13	-10
4	12	20	-8
5	15	29	-14
6	16	32	-16
7	17	23	-6
8	19	20	-1
9	23	25	-2
10	24	15	9
11	32	30	2

Subject #	Score 1	Score 2	X-Y
1	3	20	-17
2	3	13	-10
3	3	13	-10
4	12	20	-8
5	15	29	-14
6	16	32	-16
7	17	23	-6
8	19	20	-1
9	23	25	-2
10	24	15	9
11	32	30	2
		SUM:	-73



# Paired t-test

Subject #	Score 1	Score 2	X-Y	(X-Y) <sup>2</sup>
1	3	20	-17	289
2	3	13	-10	100
3	3	13	-10	100
4	12	20	-8	64
5	15	29	-14	196
6	16	32	-16	256
7	17	23	-6	36
8	19	20	-1	1
9	23	25	-2	4
10	24	15	9	81
11	32	30	2	4
		<b>SUM:</b>	<b>-73</b>	

Subject #	Score 1	Score 2	X-Y	(X-Y) <sup>2</sup>
1	3	20	-17	289
2	3	13	-10	100
3	3	13	-10	100
4	12	20	-8	64
5	15	29	-14	196
6	16	32	-16	256
7	17	23	-6	36
8	19	20	-1	1
9	23	25	-2	4
10	24	15	9	81
11	32	30	2	4
		<b>SUM:</b>	<b>-73</b>	<b>1131</b>

# Paired t-test

Subject #	Score 1	Score 2	X-Y	(X-Y)^2
1	3	20	-17	289
2	3	13	-10	100
3	3	13	-10	100
4	12	20	-8	64
5	15	29	-14	196
6	16	32	-16	256
7	17	23	-6	36
8	19	20	-1	1
9	23	25	-2	4
10	24	15	9	81
11	32	30	2	4
		<b>SUM:</b>	<b>-73</b>	<b>1131</b>

$$\bar{d} = \frac{\sum \text{Difference}}{n} = \frac{-73}{11} = -6.64$$

$$s_d = 8.04$$

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$$

$$\text{Standard error} = \frac{8.04}{\sqrt{11}} = 2.92$$

$$t = \frac{-6.64}{2.92} = -2.74$$

# t - Table

two-tails	0.2	0.1	0.05	0.02	0.01	0.002	0.001
DF							
1	3.078	6.314	12.706	31.821	63.656	318.289	636.578
2	1.886	2.92	4.303	6.965	9.925	22.328	31.6
3	1.638	2.353	3.182	4.541	5.841	10.214	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.61
5	1.476	2.015	2.571	3.365	4.032	5.894	6.869
6	1.44	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.86	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.25	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.93	4.318
13	1.35	1.771	2.16	2.65	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.14
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.12	2.583	2.921	3.686	4.015

# Decision on null Hypothesis

- Recap of Values:

- t-statistic:  $-2.74$

- Critical value:  $2.228$  (for a two-tailed test with 10 degrees of freedom and a 95% confidence level)

- Decision Rule:

- If the absolute value of the t-statistic is greater than the critical value, we can reject the null hypothesis and conclude that the difference is significant.

- If the absolute value of the t-statistic is less than the critical value, we fail to reject the null hypothesis and conclude that the difference is not significant.

- Comparison:

- $|t\text{-statistic}| = |-2.74| = 2.74$

- Critical value =  $2.228$

- Since  **$2.74 > 2.228$** , we can **reject the null hypothesis**.



# References

- <https://realpython.com/logistic-regression-python/>
  - <https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/>
- 