Robotics & Internet-of-Things | جامعة الأمير سلطان PRINCE SULTAN UNIVERSITY

CS316: INTRODUCTION TO AI AND DATA SCIENCE

CHAPTER 7
CLASSIFICATION

LECTURE 2
EVALUATE A CLASSIFICATION MODEL

Prof. Anis Koubaa

May 2024

www.riotu-lab.org

Training and Testing the Logistic Regression Model

- **Data Splitting:**
  - **Objective:** Partition the data into training and testing subsets to evaluate the model's performance on unseen data.
  - **Method:**
    - Use `train_test_split` from scikit-learn.
    - Assign 20% of the data to the test set (`test_size=0.2`).
    - Ensure consistency across runs with `random_state=42`.
- **Model Training:**
  - **Logistic Regression Setup:**
    - Initialize the Logistic Regression model.
  - **Training Process:**
    - Fit the model on the training data using `model.fit(X_train, y_train)`.
- **Key Functions Used:**
  - `train_test_split()`: Splits the dataset into separate training and testing sets.
  - `LogisticRegression()`: Creates a logistic regression model.
  - `fit()`: Trains the logistic regression model on the training data.

```python
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y,
                                                    test_size=0.2,
                                                    random_state=42)

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)
```

Evaluating Model Performance

- **Accuracy as a Performance Metric:**
  - **Purpose:** Evaluate the accuracy of the logistic regression model on the test dataset.
  - **Method:** `model.score(X_test, y_test)` calculates the accuracy, the proportion of correct predictions.
- **Code and Output:**
  - **Python Execution:**

```python
log_likelihood = model.score(X_test, y_test)  # Note: This is actu
print("Final log-likelihood:", log_likelihood)
```

  - **Interpretation:** The reported score, labeled as 'log-likelihood', is actually the accuracy of the model, which is approximately 0.869. This means the model correctly predicts the outcome for 86.9% of the test cases.
- **Key Python Function:**
  - `model.score()`: Computes the accuracy, not the log-likelihood, contrary to what the variable name suggests.

```python
# Get the final log-likelihood
log_likelihood = model.score(X_test, y_test)
print("Final log-likelihood:", log_likelihood)
```

```
Final log-likelihood: 0.8688524590163934
```

CHAPTER 3
CLASSIFICATION

CLASSIFICATION

PREDICTION WITH A LOGISTIC REGRESSION MODEL

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Making Predictions with Logistic Regression

```python
# Make predictions on the test set
y_pred = model.predict(X_test)

# Display a sample of predictions
print("Sample predictions:")
pd.DataFrame({'Actual': y_test, 'Predicted': y_pred}).head(10)
```

Sample predictions:

	Actual	Predicted
179	NO	NO
228	YES	YES
111	YES	YES
246	YES	YES
60	YES	YES
9	YES	YES
119	YES	YES
223	YES	YES
268	YES	NO
33	NO	YES

- **Model Prediction:**
  - **Objective:** Use the trained logistic regression model to make predictions on the test dataset.
  - **Process:**
    - Predict using `model.predict(X_test)`, which classifies each instance in the test set based on the learned parameters.
- **Displaying Predictions:**
  - **Sample Output:**
    - Combine actual and predicted labels in a DataFrame to compare results visually.
    - **Code Snippet:**

```python
y_pred = model.predict(X_test)
sample_predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred}).head
print("Sample predictions:", sample_predictions)
```

- **Key Functions Used:**
  - `model.predict()`: Generates label predictions for the given input data.
  - `pd.DataFrame()`: Constructs a DataFrame from the actual and predicted labels for easy comparison.

CHAPTER 3
CLASSIFICATION

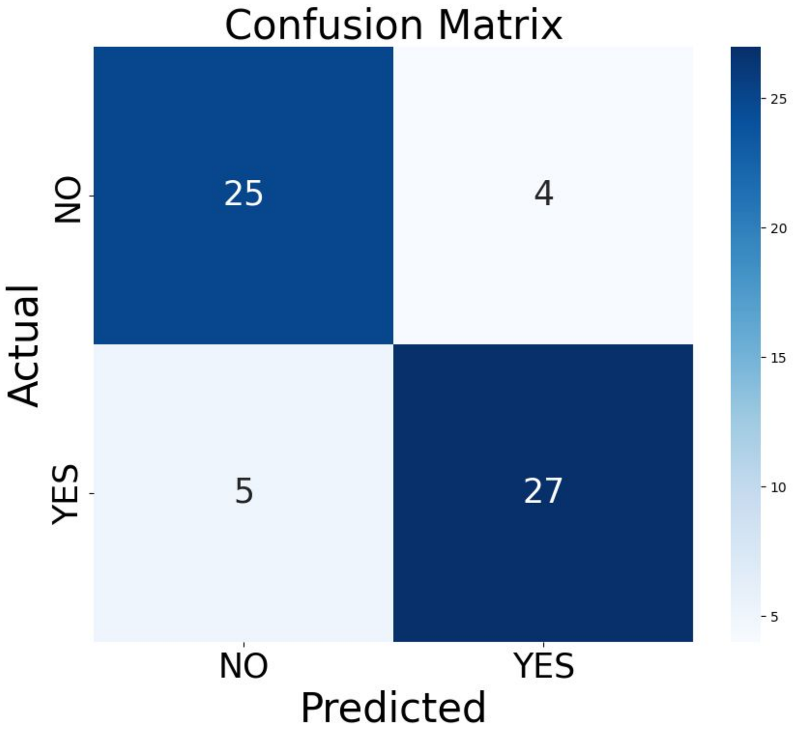
CLASSIFICATION

RESULTS INTERPRETATION AND INSIGHTS

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Confusion Matrix



Components of a Confusion Matrix

Components of a Confusion Matrix

1. True Positives (TP):

- **Definition**: The number of positive cases correctly identified as positive.
- **Explanation**: Represents accurate predictions where the actual class and predicted class both are positive.

2. True Negatives (TN):

- **Definition**: The number of negative cases correctly identified as negative.
- **Explanation**: Represents accurate predictions where the actual class and predicted class both are negative.

Components of a Confusion Matrix

3. False Positives (FP):

- **Also Known as Type I Error.**
- **Definition:** The number of negative cases incorrectly identified as positive.
- **Explanation:** Represents errors where the model incorrectly predicts the positive class.

4. False Negatives (FN):

- **Also Known as Type II Error.**
- **Definition:** The number of positive cases incorrectly identified as negative.
- **Explanation:** Represents errors where the model fails to predict the positive class when it is actually positive.

Classification Evaluation Metrics

1. Accuracy:

- **Formula:** $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
- **Explanation:** The proportion of true results (both true positives and true negatives) among the total number of cases examined.
- **Interpretation:** Indicates overall effectiveness of the model across all classifications.

2. Precision (Positive Predictive Value):

- **Formula:** $\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
- **Explanation:** The ratio of correct positive predictions to the total predicted positives.
- **Interpretation:** Useful for evaluating the cost of false positives.

Classification Evaluation Metrics

3. Recall (Sensitivity or True Positive Rate):

- **Formula**: $\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
- **Explanation**: The ratio of correct positive predictions to the actual positives.
- **Interpretation**: Critical for situations where missing a positive is significantly worse than a false positive.

4. F1 Score:

- **Formula**: $\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- **Explanation**: The harmonic mean of precision and recall, providing a balance between them.
- **Interpretation**: Best metric when seeking a balance between Precision and Recall.

Classification Evaluation Metrics

5. Specificity (True Negative Rate):

- **Formula**: $\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$
- **Explanation**: The proportion of actual negatives that are correctly identified.
- **Interpretation**: Important in cases where it is crucial to capture as many negatives as possible (e.g., screening tests).

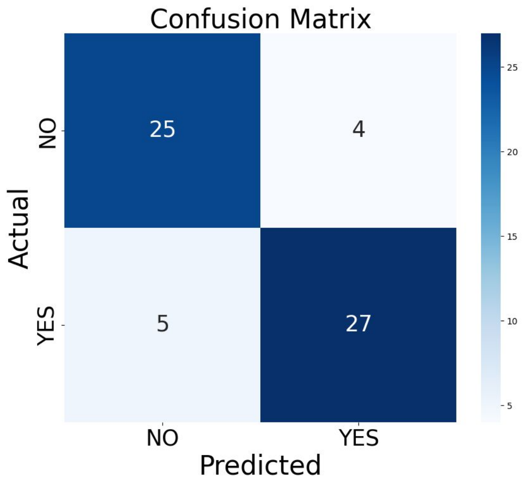
6. ROC Curve and AUC:

- **ROC Curve**: Graphical plot of the true positive rate (Recall) against the false positive rate (1 - Specificity) for different threshold settings.
- **AUC**: Area Under the ROC Curve.
- **Explanation**: AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.
- **Interpretation**: Higher AUC values indicate a better performing model, capable of differentiating between the classes effectively.

Interpretation of Heart Disease Classification Results

- **Classification Report Overview**
  - **Precision**:
    - 'NO': 0.83 - 83% of the predictions for non-disease cases were correct.
    - 'YES': 0.84 - 84% of the predictions for disease cases were correct.
  - **Recall**:
    - 'NO': 0.83 - Of all actual non-disease cases, 83% were correctly identified.
    - 'YES': 0.84 - Of all actual disease cases, 84% were correctly identified.
  - **F1-Score**:
    - 'NO': 0.83
    - 'YES': 0.84
    - Indicates balanced precision and recall for both classes.
- **Accuracy of the Model**
  - **0.84 (84%)**: Reflects that the overall predictions (both 'YES' and 'NO') are correct 84% of the time.

In a classification report, "support" refers to the number of actual occurrences of each class in the dataset used for testing the model.



Classification Report:

	precision	recall	f1-score	support
NO	0.83	0.83	0.83	29
YES	0.84	0.84	0.84	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

CS316: INTRODUCTION TO AI AND DATA SCIENCE

CHAPTER 7
CLASSIFICATION

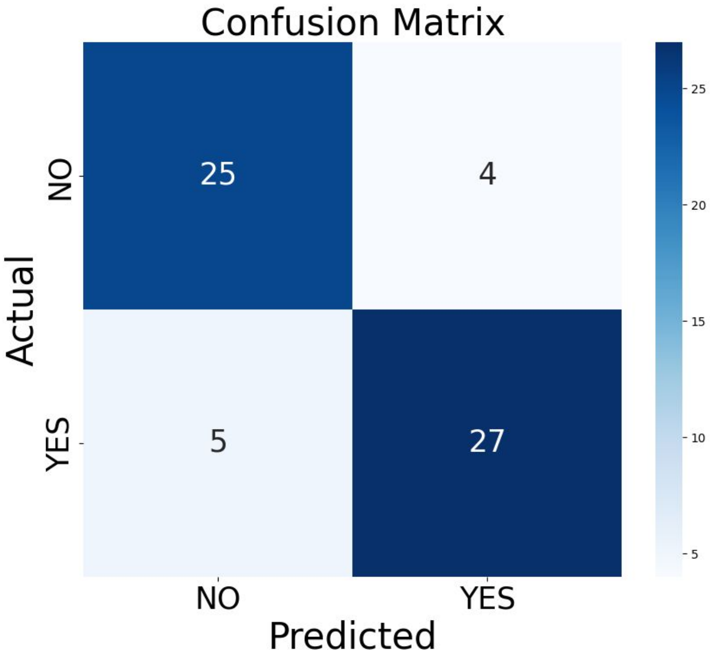
LECTURE 3

EVALUATE A CLASSIFICATION MODEL
ROC and AUC

Prof. Anis Koubaa

Nov 2024

Technical Insights and Optimization Strategies

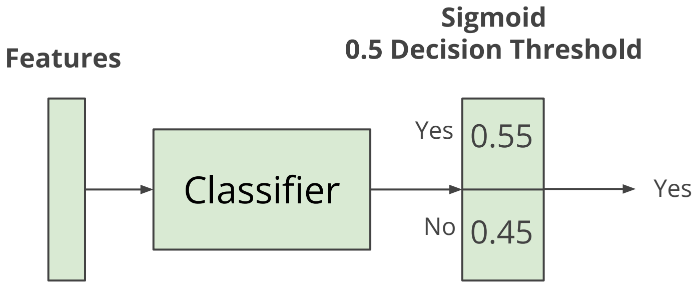


Classification Report:

	precision	recall	f1-score	support
NO	0.83	0.83	0.83	29
YES	0.84	0.84	0.84	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

Limitation of the Classification Report

- **Limitations of Classification Metrics**
    - **Precision and Recall Limitations:**
        - Dependent on a specific **decision threshold.**
        - Do not provide insight into classification performance across varying thresholds.
    - **Inadequate for Imbalanced Datasets:**
        - Metrics can be biased towards the **majority class**, misleading model evaluation.



Classification Report:

	precision	recall	f1-score	support
NO	0.83	0.83	0.83	29
YES	0.84	0.84	0.84	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

What is ROC?

ROC Curve (Receiver Operating Characteristic Curve):

- The ROC curve is a graphical representation of a classification model's performance at different threshold settings.
- It plots **True Positive Rate (TPR)** against **False Positive Rate (FPR)** at various threshold values.
- **True Positive Rate (Sensitivity):**

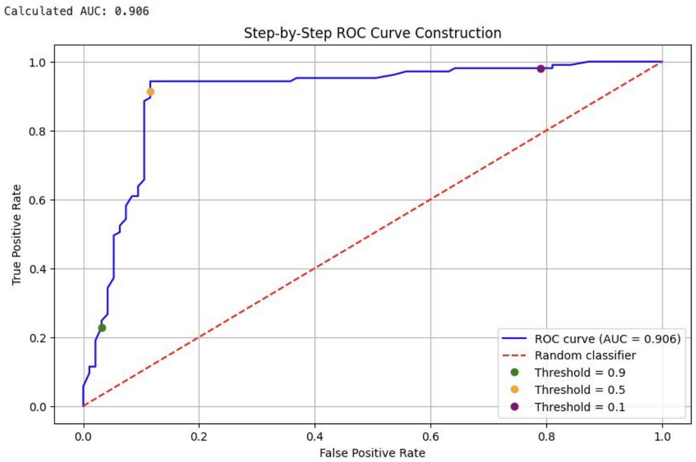
$$TPR = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Measures how well the model identifies positive cases.

- **False Positive Rate:**

$$FPR = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Measures how often the model incorrectly identifies a negative case as positive.

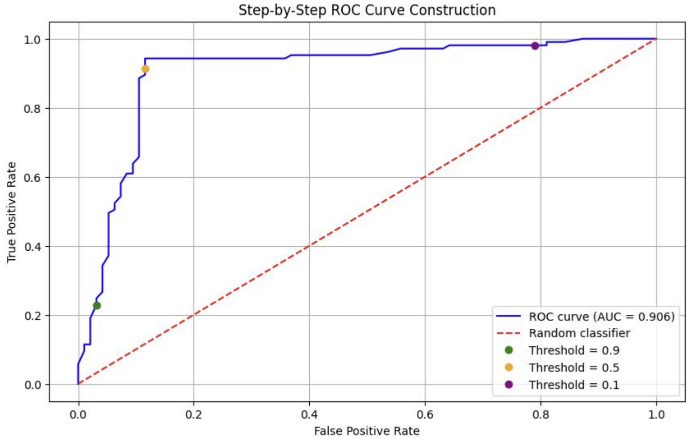


What is AUC?

AUC (Area Under the Curve):

- The **AUC** is the area under the ROC curve, summarizing the model's ability to distinguish between classes.
- It ranges from 0 to 1:
  - **1.0**: Perfect classifier.
  - **0.5**: Random guess (no discrimination ability).
  - **< 0.5**: Worse than random (usually a sign of model issues).

Calculated AUC: 0.906



How to Interpret ROC AUC?

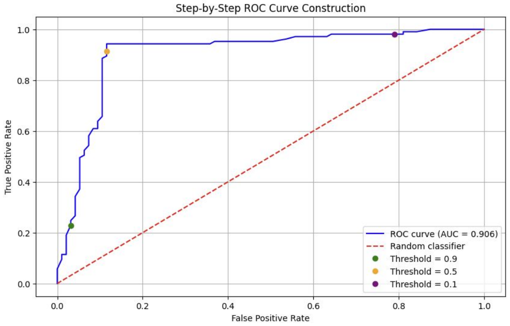
How to Interpret ROC AUC?

- **Higher AUC means better performance.** It indicates that the model is good at distinguishing between positive and negative cases.
- For example:
  - An AUC of **0.9** means that in 90% of randomly chosen cases, the model ranks a positive instance higher than a negative one.

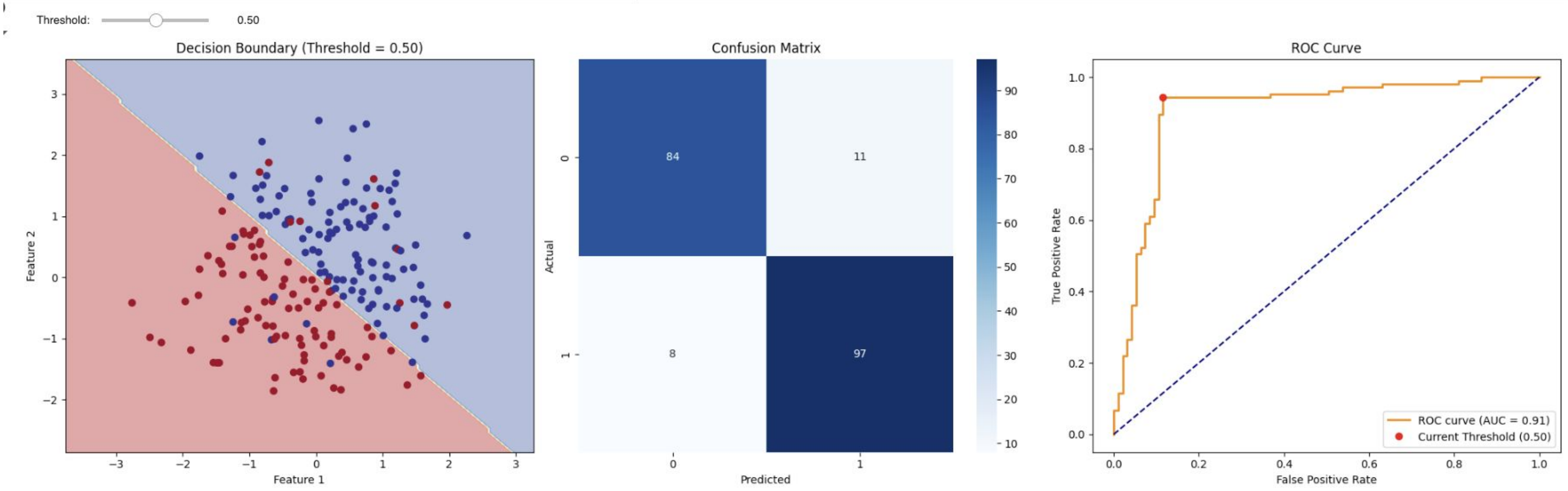
Why Use ROC AUC?

- It is a threshold-independent metric, meaning it evaluates the model across all possible thresholds.
- It is especially useful for imbalanced datasets because it focuses on the model's ability to rank predictions correctly, rather than its accuracy.

Calculated AUC: 0.906



How to Interpret ROC AUC?




```
Performance Metrics at Current Threshold:
----------------------------------------
True Positive Rate (Sensitivity) 0.924
False Positive Rate              0.116
Precision                        0.898
Accuracy                         0.905
Specificity                      0.884
```

How to plot a ROC?

Input

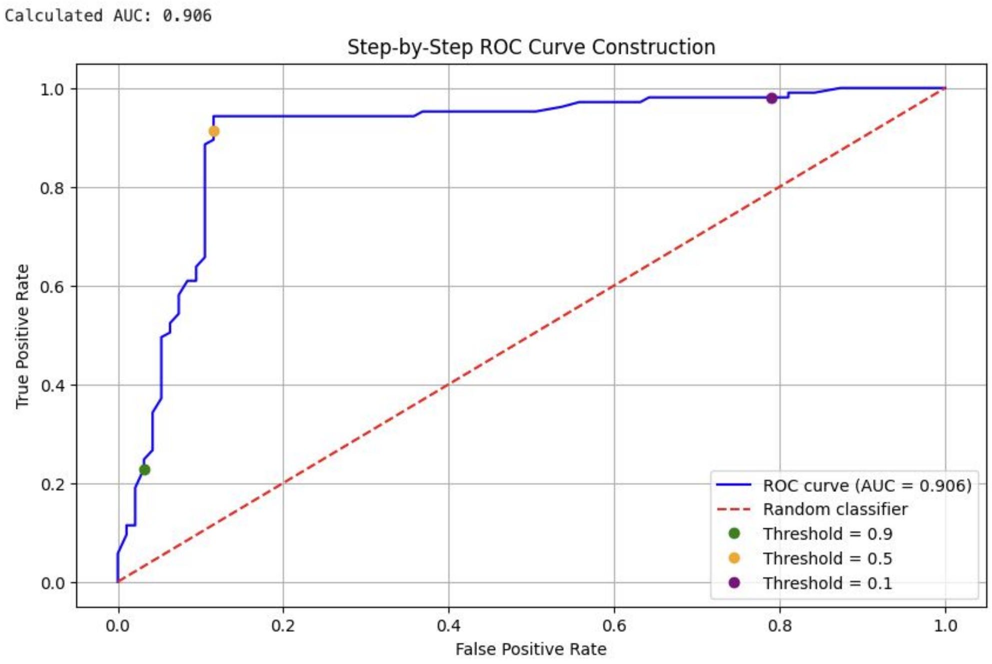
- **True labels** (0/1)
- **Predicted probabilities** (0.0 to 1.0)

Algorithm Steps

1. **Create 100 thresholds** from 1.0 to 0.0
2. **For each threshold:**
   - **Convert probabilities to predictions:**
     - If probability ≥ threshold → predict 1
     - If probability < threshold → predict 0
   - **Count:**
     - **True Positives (TP)** = prediction 1, actual 1
     - **True Negatives (TN)** = prediction 0, actual 0
     - **False Positives (FP)** = prediction 1, actual 0
     - **False Negatives (FN)** = prediction 0, actual 1
   - **Calculate rates:**
     - **TPR** = TP / (TP + FN)
     - **FPR** = FP / (FP + TN)
   - **Store point** (FPR, TPR)
3. **Calculate AUC** using stored points

Output

- **ROC Curve:** Plot of FPR vs TPR
- **AUC Score:** Area under the curve

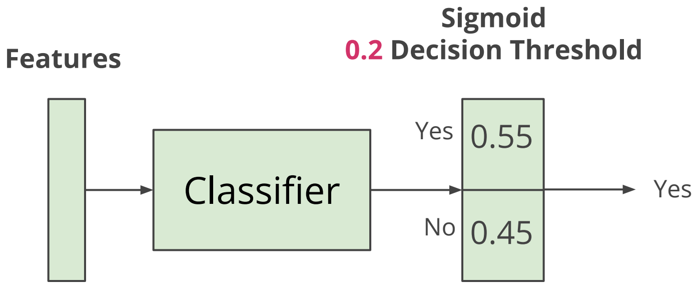


Limitation of the Classification Report

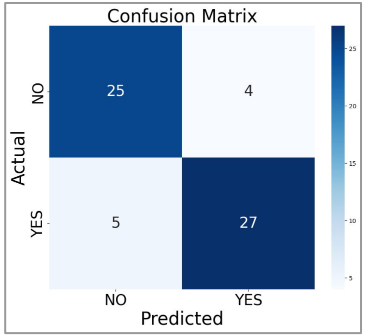
Classification Report:

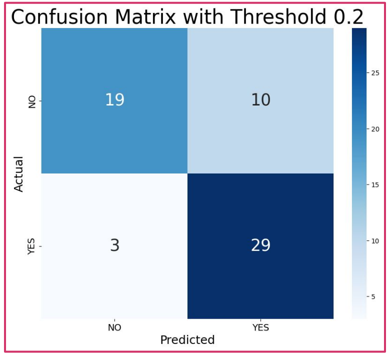
	precision	recall
NO	0.83	0.83
YES	0.84	0.84

- **Limitations of Classification Metrics**
  - **Precision and Recall Limitations:**
    - Dependent on a specific **decision threshold.**
      - *Precision drops from* ***0.85 to 0.74*** *when the decision threshold is lowered from* ***0.5 to 0.2****, illustrating sensitivity to threshold settings.*
    - Do not provide insight into classification performance across varying thresholds.
  - **Inadequate for Imbalanced Datasets:**
    - Metrics can be biased towards the **majority class**, misleading model evaluation.




```
New Threshold - Accuracy: 0.79
New Threshold - Precision: 0.74
New Threshold - Recall: 0.91
```

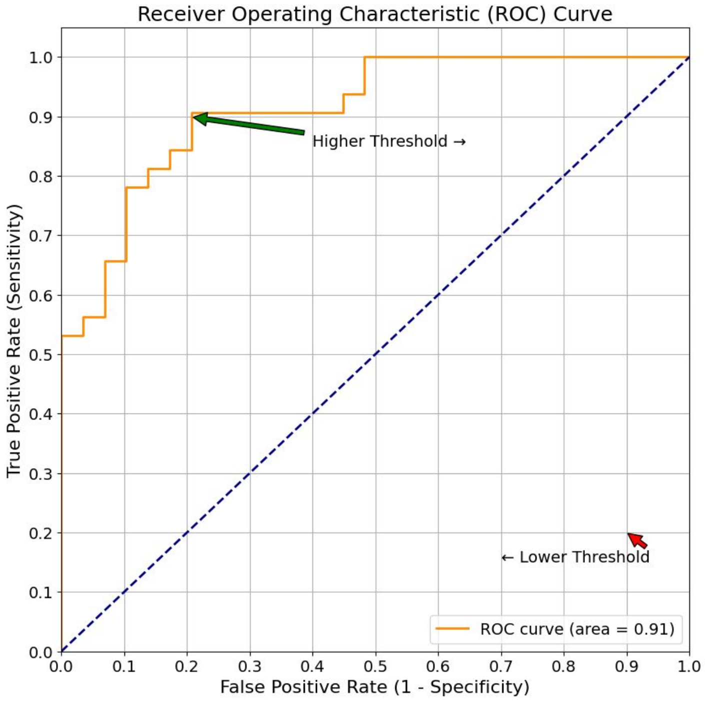




END OF LECTURE

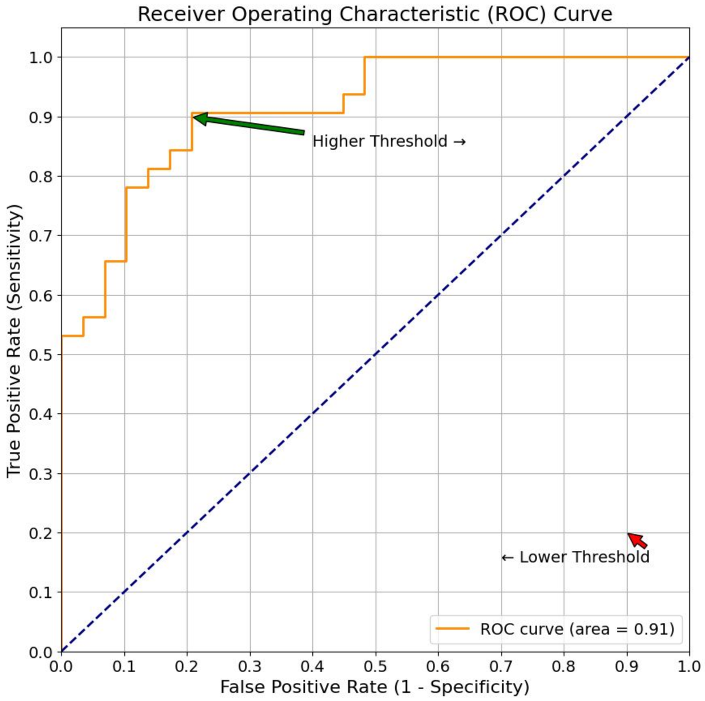
ROC Analysis in Model Evaluation

- **Understanding the ROC Curve**
  - **Curve Overview**: The orange ROC curve shows how sensitivity (TPR) and 1-specificity (FPR) vary at different threshold levels.
  - **Threshold Movement**: As the threshold for classifying a patient as having the disease is lowered (moving from right to left), both the sensitivity and false positive rate increase.
- **ROC Curve Insights**
  - **Steep Initial Ascent**: Indicates rapid gain in sensitivity with minimal increase in false positives, important for early disease detection.
  - **Curve Plateau**: Shows diminishing returns in sensitivity, highlighting limits in model performance as threshold decreases.
- **AUC Value: 0.91**
  - Indicates excellent discrimination ability, essential for reliable disease diagnosis in healthcare.
  - Suggests that the model correctly ranks a positive instance over a negative one in 91% of comparisons.
- **Healthcare Application**
  - **Optimal Threshold Setting**: Helps determine the most effective balance between detecting disease cases (sensitivity) and minimizing false alarms (specificity).
  - **Model Selection**: Prefer models with higher AUC for critical diagnostic applications to ensure accurate patient outcomes.



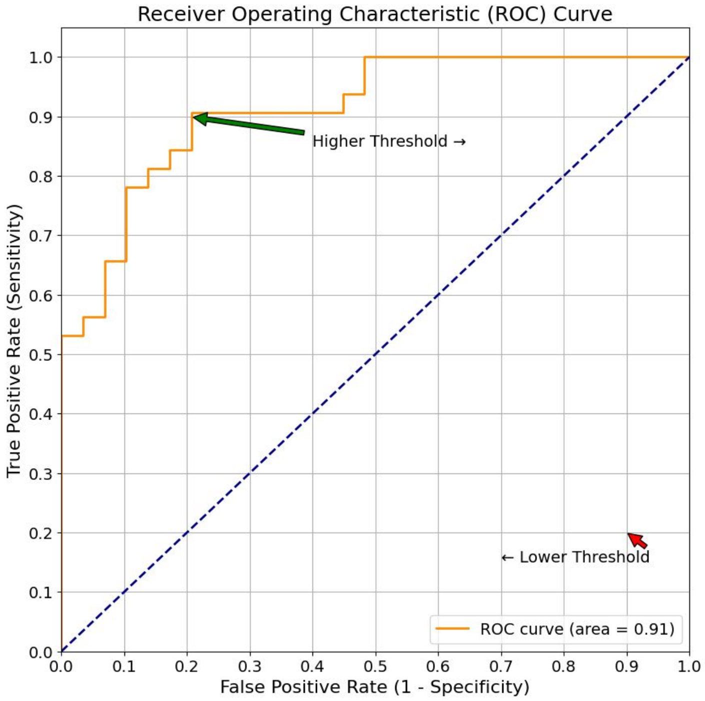
ROC Analysis in Model Evaluation

- **Motivation: Overcoming Limitations**
  - Traditional metrics are threshold-specific and fail to depict model performance comprehensively, especially under variable conditions and class imbalances.
- **ROC Curve Defined**
  - **ROC Curve**: Plots True Positive Rate (TPR) against False Positive Rate (FPR) to show model performance across thresholds.
  - **Purpose**: Evaluates the ability to distinguish between classes under varying conditions.
- **Mathematical Formulation**
  - **True Positive Rate (TPR)**: $\text{Sensitivity} = \frac{TP}{TP+FN}$
  - **False Positive Rate (FPR)**: $1 - \text{Specificity} = \frac{FP}{FP+TN}$
- **Utility as a Classification Metric**
  - Enables comprehensive evaluation by illustrating the trade-off between sensitivity and false alarms, crucial for determining the most effective operational threshold.



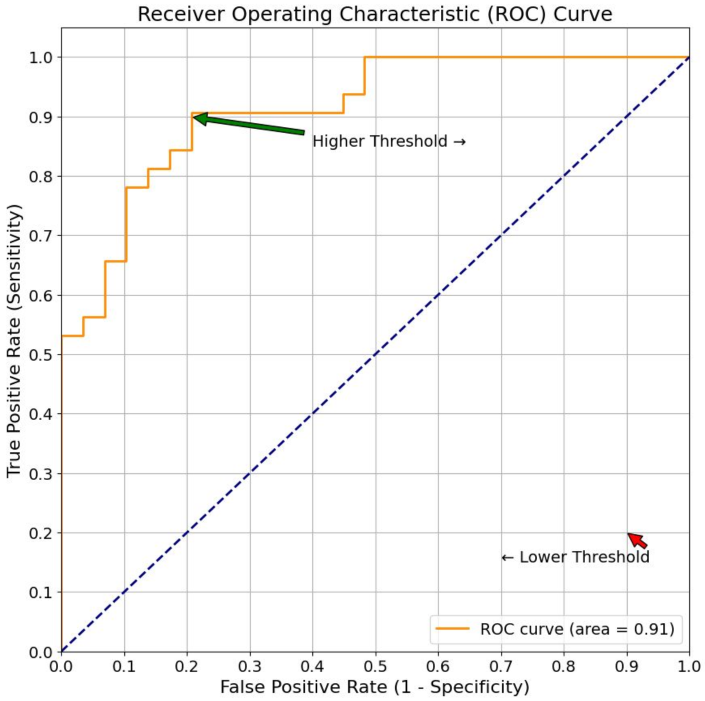
Plotting the ROC Curve: An Intuitive Guide

- **Objective**: Illustrate a model's discrimination ability across various thresholds.
- **Algorithm Steps**:
  1. **Initialize**: Set the highest threshold (typically 1) where no positives are predicted; start at $(0, 0)$ on the ROC plot.
  2. **Calculate for Each Threshold**:
     - Decrement threshold from 1 to 0.
     - **True Positive Rate (TPR)**: $\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
     - **False Positive Rate (FPR)**: $\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$
     - Compute $\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN}$ based on each threshold.
  3. **Plot**: For each threshold, plot (FPR, TPR) to trace the ROC curve from the bottom left (most conservative) to the top right (least conservative).
- **Evaluate**:
  - Closer the curve approaches the top left corner, the better the classifier.
  - The area under the curve (AUC) measures overall effectiveness.



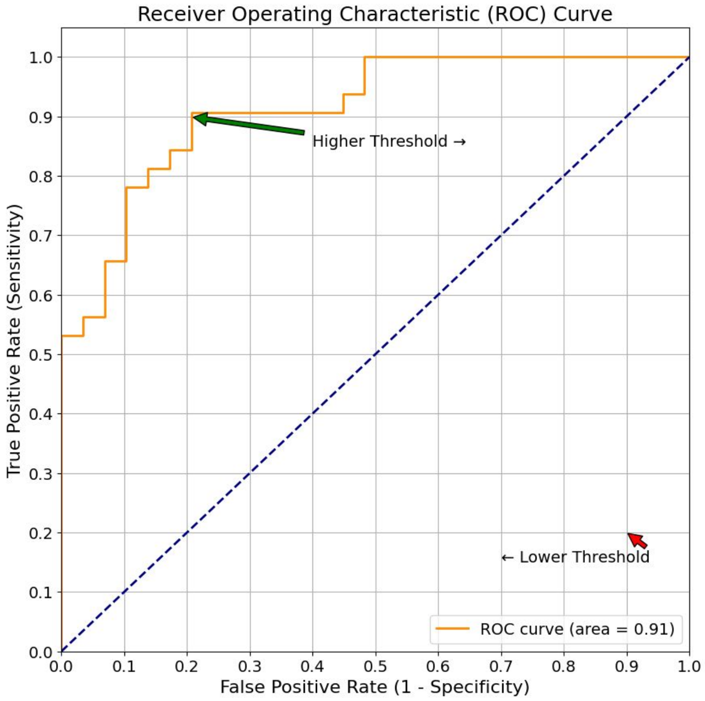
Interpreting the ROC Curve Intuitively

- **Key Aspects of ROC Curve Interpretation**
  - **Above the Diagonal**: Indicates better-than-random performance. The higher and closer to the top left corner, the better the model's sensitivity and specificity.
  - **Shape and Direction**: A steep initial ascent suggests high sensitivity for a low false positive rate—ideal in many practical applications.
  - **Area Under Curve (AUC)**: A direct measure of model accuracy; higher AUC values (closer to 1.0) signify better overall model performance.



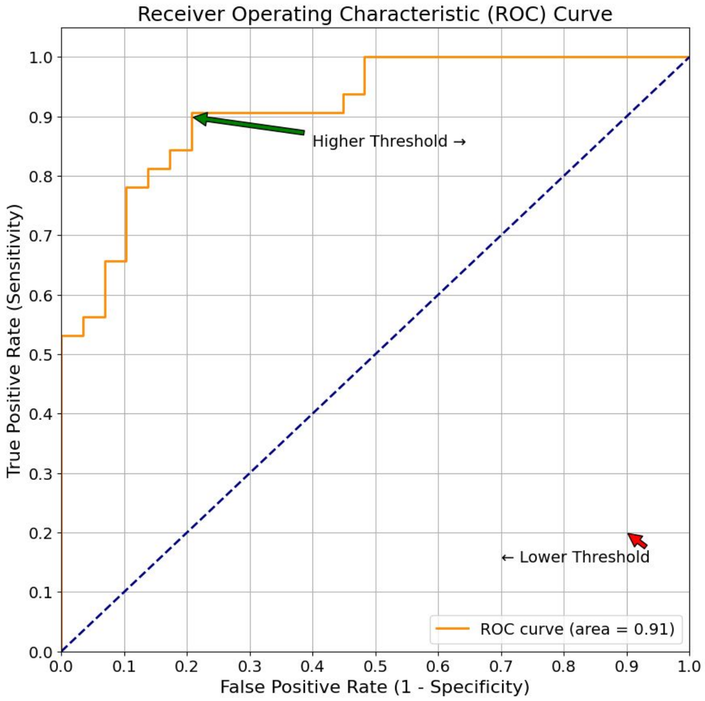
Understanding the Diagonal in ROC Curve Plots

- **Baseline Performance**
  - The diagonal line represents a random classifier, serving as a baseline for comparison.
- **Indication of No Discrimination**
  - Slope of 1 (45-degree angle) indicates no ability to distinguish between classes; akin to flipping a coin.
- **Reference for Model Evaluation**
  - Effective classifiers exhibit ROC curves above this line, showing superior discrimination.
  - Curves below suggest performance worse than random, signaling model issues.



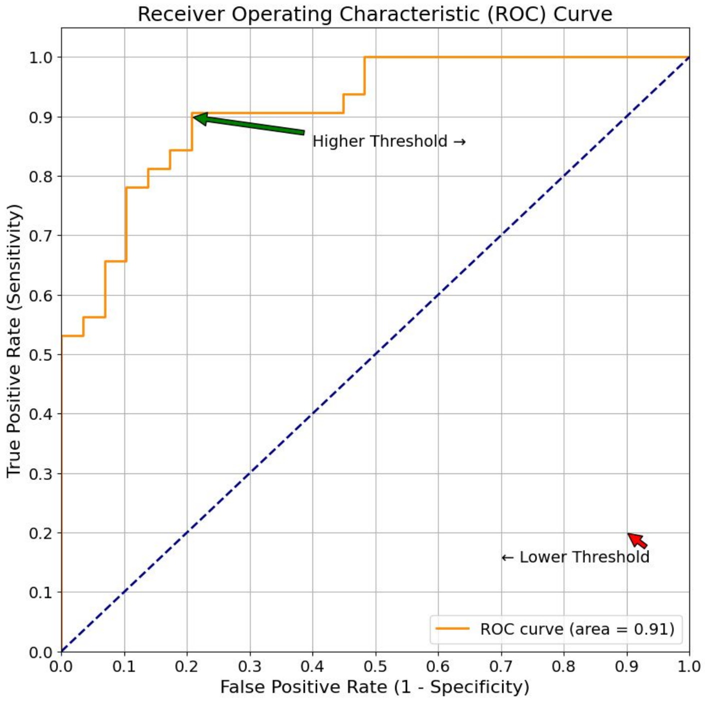
Area Under the Curve

- **What is AUC?**
  - AUC, or Area Under the ROC Curve, quantifies a model's ability to correctly classify positives and negatives at various threshold levels.
- **Why is AUC Important?**
  - **Comprehensive Measure**: It evaluates model performance over all possible classification thresholds, not just one.
  - **Model Comparison**: Simplifies comparing the effectiveness of different models by providing a single value ranging from 0.5 (no discrimination) to 1.0 (perfect discrimination).
- **Practical Use**
  - **Optimal Selection**: A higher AUC indicates a model with better overall predictive accuracy, essential in applications like medical diagnosis or fraud detection.



Area Under the Curve

- **Definition of AUC**
  - Measures the area underneath the ROC curve; evaluates a model's ability to distinguish between classes across all thresholds.
- **Significance of AUC**
  - **Scalar Value Interpretation**:
    - **1.0**: Perfect discrimination.
    - **0.5**: No better than random.
    - **<0.5**: Worse than random, indicates potential model issues.
  - **Advantages**:
    - Threshold-independent, providing a comprehensive overview.
    - Enables straightforward comparisons between different models.



Why Accuracy is Not Sufficient?

```
True Negatives: 25
False Positives: 4
False Negatives: 5
True Positives: 27
Interpretation:
False Positives (Type I Error): 4 - Patients incorrectly diagnosed with heart disease.
False Negatives (Type II Error): 5 - Patients with heart disease missed by the model.
```

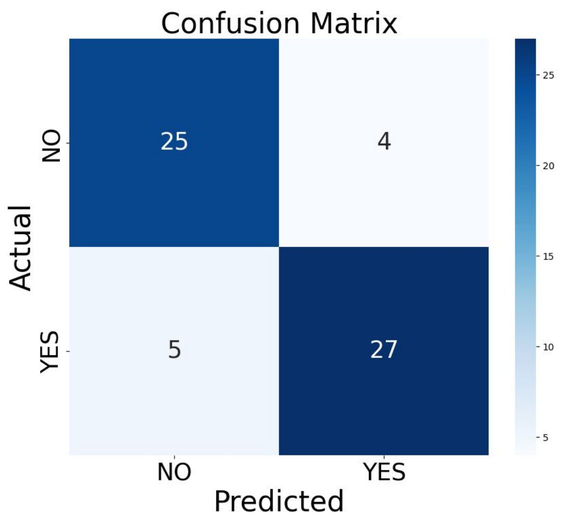
```python
# Make predictions on the test set
y_pred = model.predict(X_test)

# Display a sample of predictions
print("Sample predictions:")
pd.DataFrame({'Actual': y_test, 'Predicted': y_pred}).head(10)
```

Sample predictions:

	Actual	Predicted
179	NO	NO
228	YES	YES
111	YES	YES
246	YES	YES
60	YES	YES
9	YES	YES
119	YES	YES
223	YES	YES
268	YES	NO
33	NO	YES

Confusion Matrix

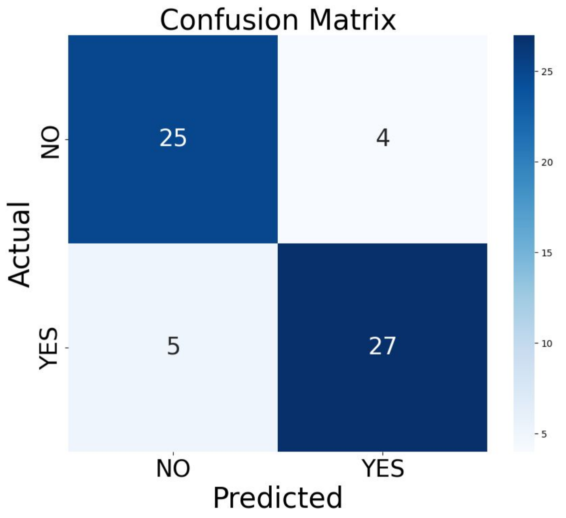
- **Confusion Matrix Components**
  - **True Negatives (TN):** 25 patients correctly identified as not having heart disease.
  - **False Positives (FP)**, Type I Error: 4 patients incorrectly diagnosed with heart disease.
  - **False Negatives (FN)**, Type II Error: 5 patients with heart disease missed by the model.
  - **True Positives (TP):** 27 patients correctly identified as having heart disease.
- **Implications of Misclassification**
  - **False Positives (FP):** Unnecessary treatment and emotional distress.
  - **False Negatives (FN):** Missed diagnoses, potentially leading to a lack of necessary medical intervention.
- **Significance**
  - Highlights the importance of considering more than just accuracy in evaluating model performance.
  - Misclassifications can have serious real-world consequences, particularly in medical settings.



- **Impact of Errors**
  - **Type I Error (FP):** May lead to unnecessary anxiety and treatment.
  - **Type II Error (FN):** Critical as it may result in lack of necessary treatment.

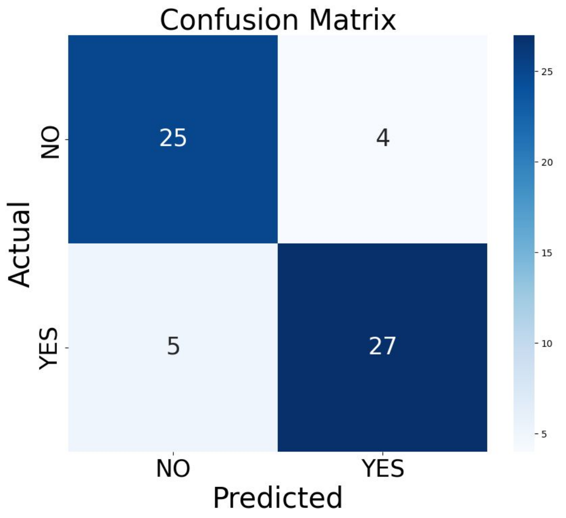
Accuracy in Model Evaluation

- **General Definition of Accuracy**
  - **Formal Expression:**
    - $\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$
- **Accuracy Using Confusion Matrix**
  - Incorporates True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).
  - **Mathematical Formalism:**
    - $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
- **Implications of Using Accuracy**
  - Provides an overall measure of model performance.
  - Does not highlight the impact of misclassification types (FP, FN).
  - Importance in applications: Misclassification can lead to serious consequences, particularly in fields like healthcare.



Precision in Model Evaluation

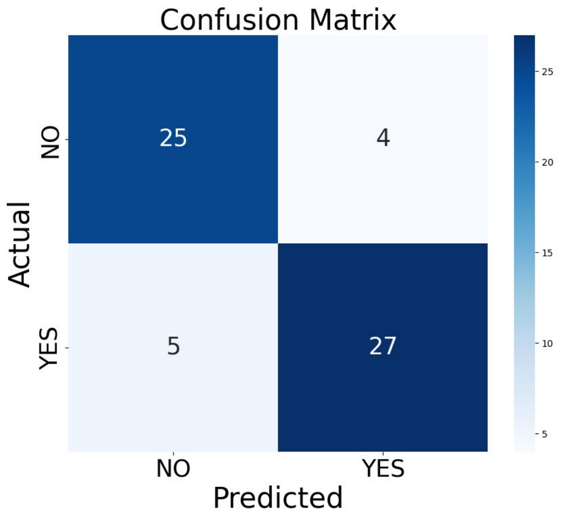
- **General Definition of Precision**
  - **Formal Expression**:
    - $\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}}$
- **Contrasting Precision with Accuracy**
  - **Precision**: Focuses specifically on the correctness of positive predictions.
  - **Accuracy**: Measures overall correctness of all predictions (both positive and negative).
  - **Value of Precision**: Particularly important in scenarios where the cost of a false positive is high.
- **Precision in Healthcare**
  - **Example**: In diagnosing a disease like heart disease, precision ensures that most patients diagnosed positively truly have the disease.
  - Minimizes the risk of unnecessary treatments that could result from false positives.
- **Implications of Using Precision**
  - Emphasizes quality over quantity in positive predictions.
  - Essential for medical diagnostics where the consequence of treating a non-ill patient (FP) can be detrimental.




```
Accuracy: 0.84
Precision: 0.84
Discussion:
High Precision indicates fewer false positives, which is crucial in medical scenarios to avoid unnecessary treatments.
High Accuracy indicates the overall correctness of the model, ensuring that it performs well across all classes.
```

Recall/Sensitivity in Model Evaluation

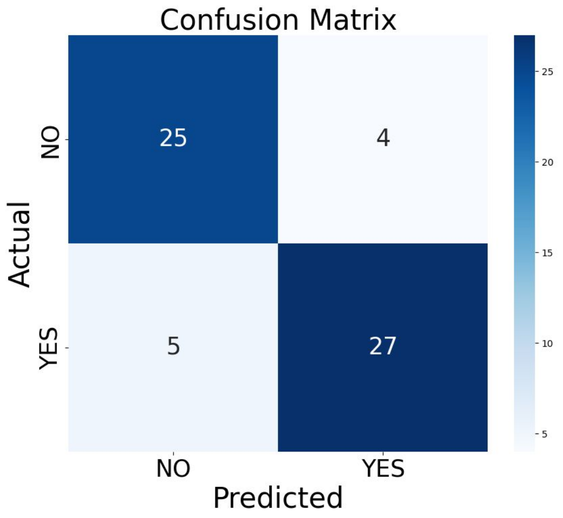
- **General Definition of Recall**
  - **Formal Expression**:
    - $\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$
- **Contrasting Recall with Accuracy and Precision**
  - **Recall**: Measures the model's ability to detect all actual positives.
  - **Accuracy**: Measures the overall correctness of both positive and negative predictions.
  - **Precision**: Measures the correctness of only the positive predictions.
  - **Value of Recall**: Critical in scenarios where missing a positive case (FN) carries significant risk.
- **Recall in Healthcare**
  - **Example**: In heart disease diagnosis, a recall of 0.84 means the model correctly identifies 84% of all actual disease cases.
  - High recall reduces the risk of missing a diagnosis, which is crucial to ensure timely and appropriate medical treatment.




```
Recall: 0.84
Accuracy: 0.84
Precision: 0.84
Discussion on Recall:
High Recall indicates that the model is good at identifying all actual positives.
This is crucial for medical diagnostics to ensure that no cases of the disease are missed, which could be life-threatening.
```

F1-Score in Model Evaluation

- **General Definition of F1-Score**
  - **Formal Expression**:
    - $\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- **Contrasting F1-Score with Accuracy, Precision, and Recall**
  - **F1-Score**: Harmonic mean of precision and recall, balancing the two metrics.
  - **Accuracy**: General measure of correctness across all predictions.
  - **Precision**: Focus on correct positive predictions only.
  - **Recall**: Focus on capturing all actual positives.
  - **Value of F1-Score**: Especially useful in conditions where an equal balance between precision and recall is crucial.
- **F1-Score in Healthcare**
  - **Example**: With an F1-Score of 0.84 in heart disease diagnosis, this metric indicates a strong balance between precision and recall, essential for effective and efficient patient care.
  - Ensures that the model is not only accurate but also balanced in terms of avoiding unnecessary treatments and not missing critical diagnoses.
- **Implications of Using F1-Score**
  - Provides a single metric that condenses the trade-offs between precision and recall.
  - Crucial in medical settings where both false positives and false negatives have serious implications.



CHAPTER 3
CLASSIFICATION

CLASSIFICATION

MODEL EVALUATION IN PYTHON

CS316: INTRODUCTION TO AI AND DATA SCIENCE

Prof. Anis Koubaa

Model Evaluation in Python – Individual Metrics

- **Using scikit-learn for Model Evaluation**
  - **Libraries and Functions**:
    - Utilize `sklearn.metrics` for comprehensive model evaluation tools.
    - Functions used: `accuracy_score`, `precision_score`, `recall_score`, `f1_score`.
- **Calculation of Metrics**
  - **Accuracy**:
    - `accuracy = accuracy_score(y_test, y_pred)`
    - Represents overall correctness of the model.
  - **Precision**:
    - `precision = precision_score(y_test, y_pred, pos_label='YES')`
    - Indicates the accuracy of positive predictions.
  - **Recall**:
    - `recall = recall_score(y_test, y_pred, pos_label='YES')`
    - Measures the model's ability to identify all relevant instances.
  - **F1 Score**:
    - `f1 = f1_score(y_test, y_pred, pos_label='YES')`
    - Harmonic mean of precision and recall, balancing both metrics.

```python
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, classification_report
# Calculate metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, pos_label='YES')
recall = recall_score(y_test, y_pred, pos_label='YES')
f1 = f1_score(y_test, y_pred, pos_label='YES')

# Print basic metrics
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
```

```
Accuracy: 0.84
Precision: 0.84
Recall: 0.84
F1 Score: 0.84
```

Classification Report & Confusion Matrix

- **Python Code for Model Evaluation**
  - Use `classification_report` and `confusion_matrix` from `sklearn.metrics` for detailed performance analysis.
- **Classification Report**
  - **Math Formulas**:
    - Precision: $\text{Precision} = \frac{TP}{TP+FP}$
    - Recall: $\text{Recall} = \frac{TP}{TP+FN}$
    - F1-Score: $\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
  - **Code**: `report = classification_report(y_test, y_pred, target_names=model.classes_)`
  - Outputs precision, recall, and F1-score for each class.
- **Confusion Matrix**
  - **Visualization with Python**:
    - **Code**:

```python
cm = confusion_matrix(y_test, y_pred, labels=model.classes_)
cm_df = pd.DataFrame(cm, index=model.classes_, columns=model.classes_)
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', annot_kws={"size": 16})
```

    - Displays predicted vs actual classifications in a heatmap format.

```python
# Generate and print the classification report
report = classification_report(y_test, y_pred, target_names=model.classes_)
print("Classification Report:")
print(report)

# Compute confusion matrix to evaluate the accuracy of a classification
cm = confusion_matrix(y_test, y_pred, labels=model.classes_)
cm_df = pd.DataFrame(cm, index=model.classes_, columns=model.classes_)

# Plotting the confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(cm_df, annot=True, fmt='d', cmap='Blues', annot_kws={"size": 16})
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted', size=18)
plt.ylabel('Actual', size=18)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.show()
```

```
Classification Report:
              precision    recall  f1-score   support

          NO       0.83      0.83      0.83        29
         YES       0.84      0.84      0.84        32

    accuracy                           0.84        61
   macro avg       0.84      0.84      0.84        61
weighted avg       0.84      0.84      0.84        61
```