Machine Learning
Lecture 2

Introduction to ML

Associate: Wafaa Shalash
Benha Fac. of Computers and AI

Autumn 2024

Performance Metrics

Performance Evaluation

- Performance indicators:
- Area under curve (AUC).
- Receiver operating characteristics (ROC) curve.
- TPR or Sensitivity (SN):= $\frac{TP}{TP+FN}$
- TNR or Specificity (SP):$\frac{TN}{TN+FP}$
- Accuracy $(ACC) = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision or PPV = $\frac{TP}{TP+Fp}$
- FDR = $\frac{FP}{FP+TP}$

Accuracy

- Accuracy shows the ratio of correct predictions to all predictions:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Number\ of\ all\ predictions}$$

Limitation: Can be misleading with imbalanced datasets (e.g., a 95% accuracy in detecting rare diseases might mean only detecting the majority class).

Precision, Recall, and F1-Score

Let’s assume class A is positive class and class B is negative class.
The key terms of confusion matrix are as follows:

- **True positive (TP)**: Predicting positive class as positive (ok)
- **False positive (FP)**: Predicting negative class as positive (not ok)
- **False negative (FN)**: Predicting positive class as negative (not ok)
- **True negative (TN)**: Predicting negative class as negative (ok)

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

	Actual	*Prediction*	*Evaluation*
TP	Positive	Positive	OK
FP	Negative	Positive	Not OK
FN	Positive	Negative	Not OK
TN	Negative	Negative	OK

Precision and Recall

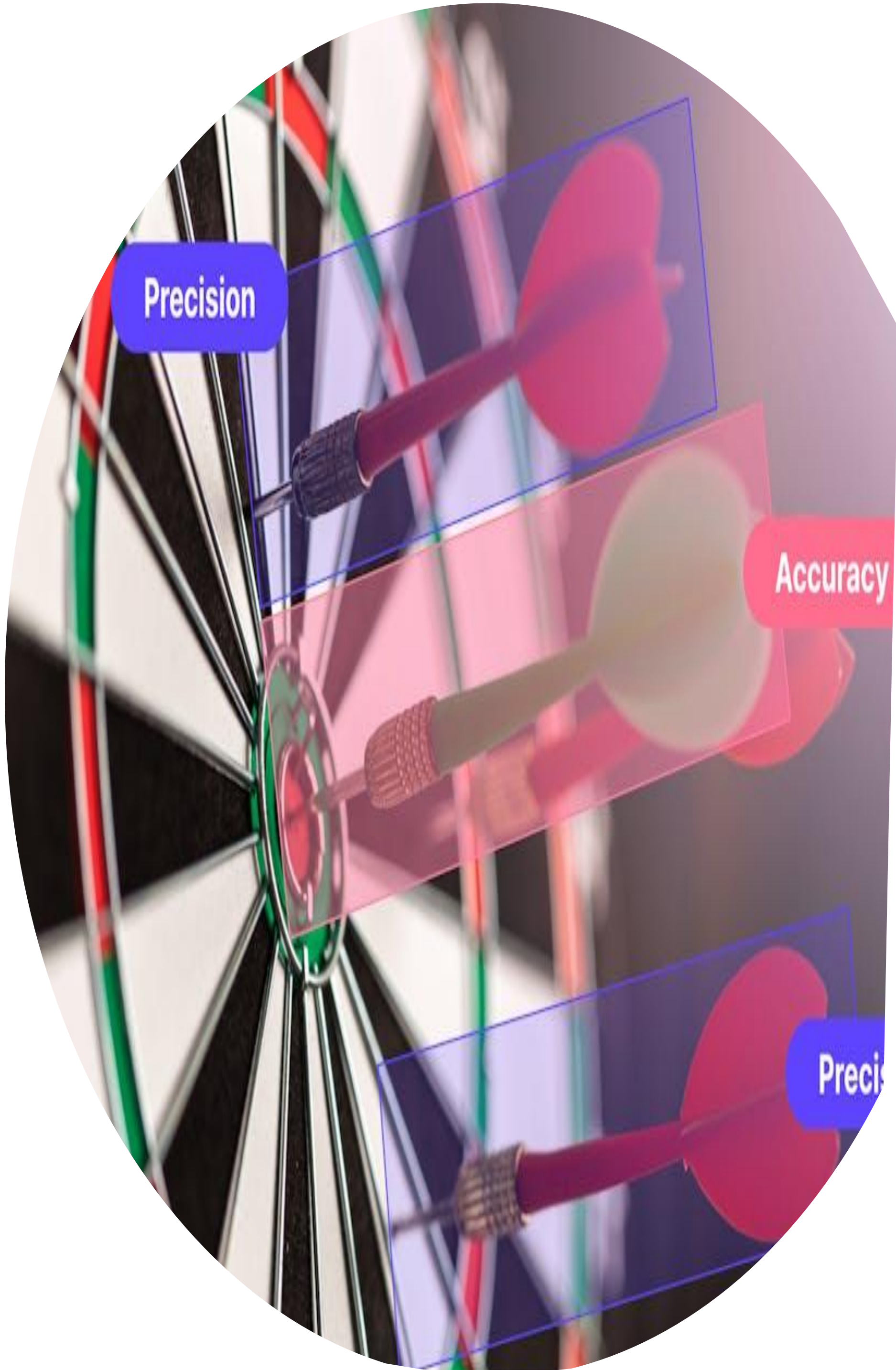
- Precision and recall metrics take the classification accuracy one step further and allow us to get a more specific understanding of model evaluation. Which one to prefer depends

- **Precision** measures how good our model is when the prediction is positive. It is the ratio of correct positive predictions to all positive **predictions**: on the task and what we aim to achieve.
- **Recall** measures how good our model is at correctly predicting positive classes. It is the ratio of correct positive predictions to all positive **classes**.

The focus of precision is **positive predictions** so it indicates how many positive predictions are true. The focus of recall is **actual positive classes** so it indicates how many of the positive classes the model is able to predict correctly.

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

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



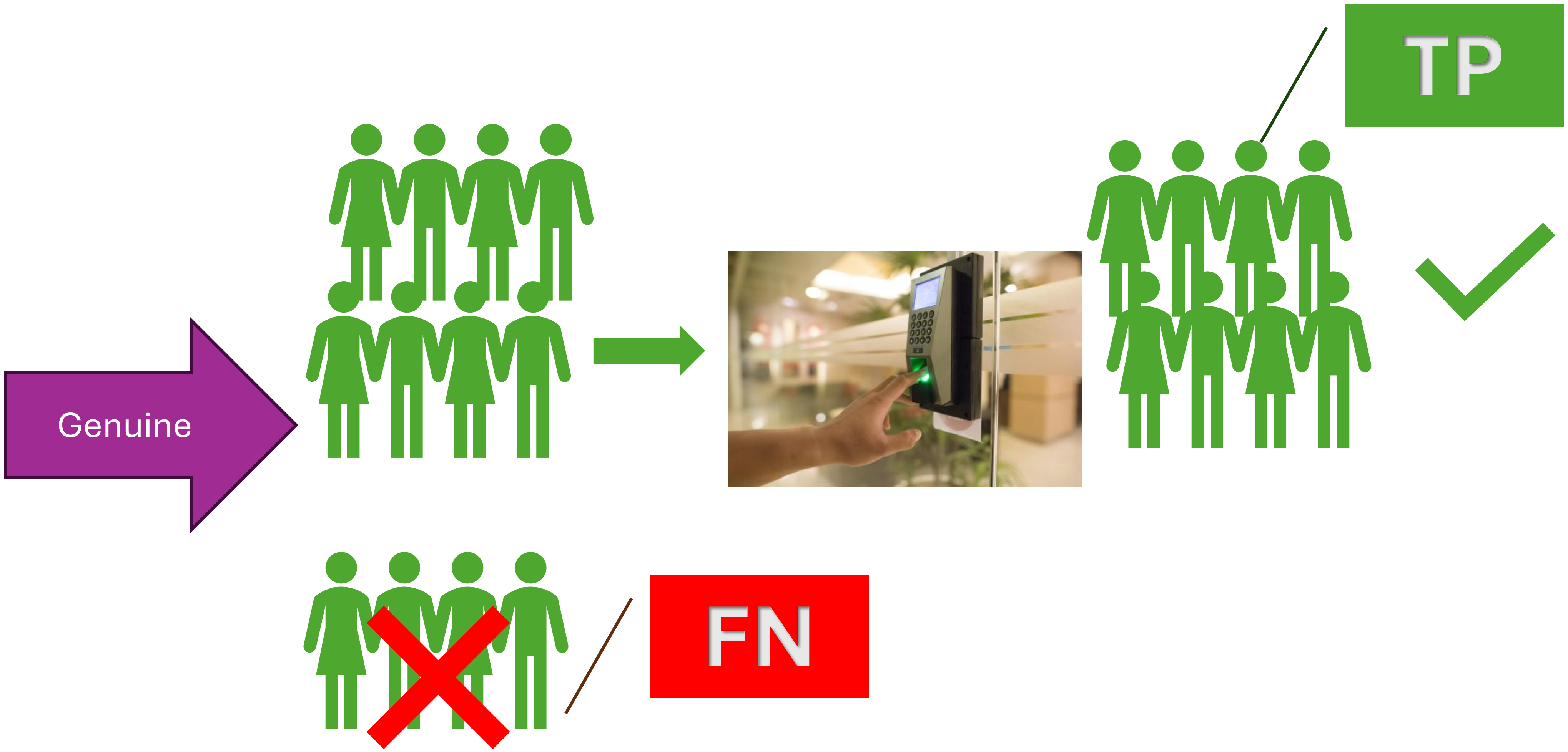
Confusion Matrix

- A confusion matrix is not a metric to evaluate a model, but it provides insight into the predictions. It is important to learn confusion matrix in order to comprehend other classification metrics such as **precision** and **recall**.
- Confusion matrix goes deeper than classification accuracy by showing the correct and incorrect (i.e. true or false) predictions on each class. In case of a binary classification task, a confusion matrix is a 2x2 matrix. If there are three different classes, it is a 3x3 matrix and so on.

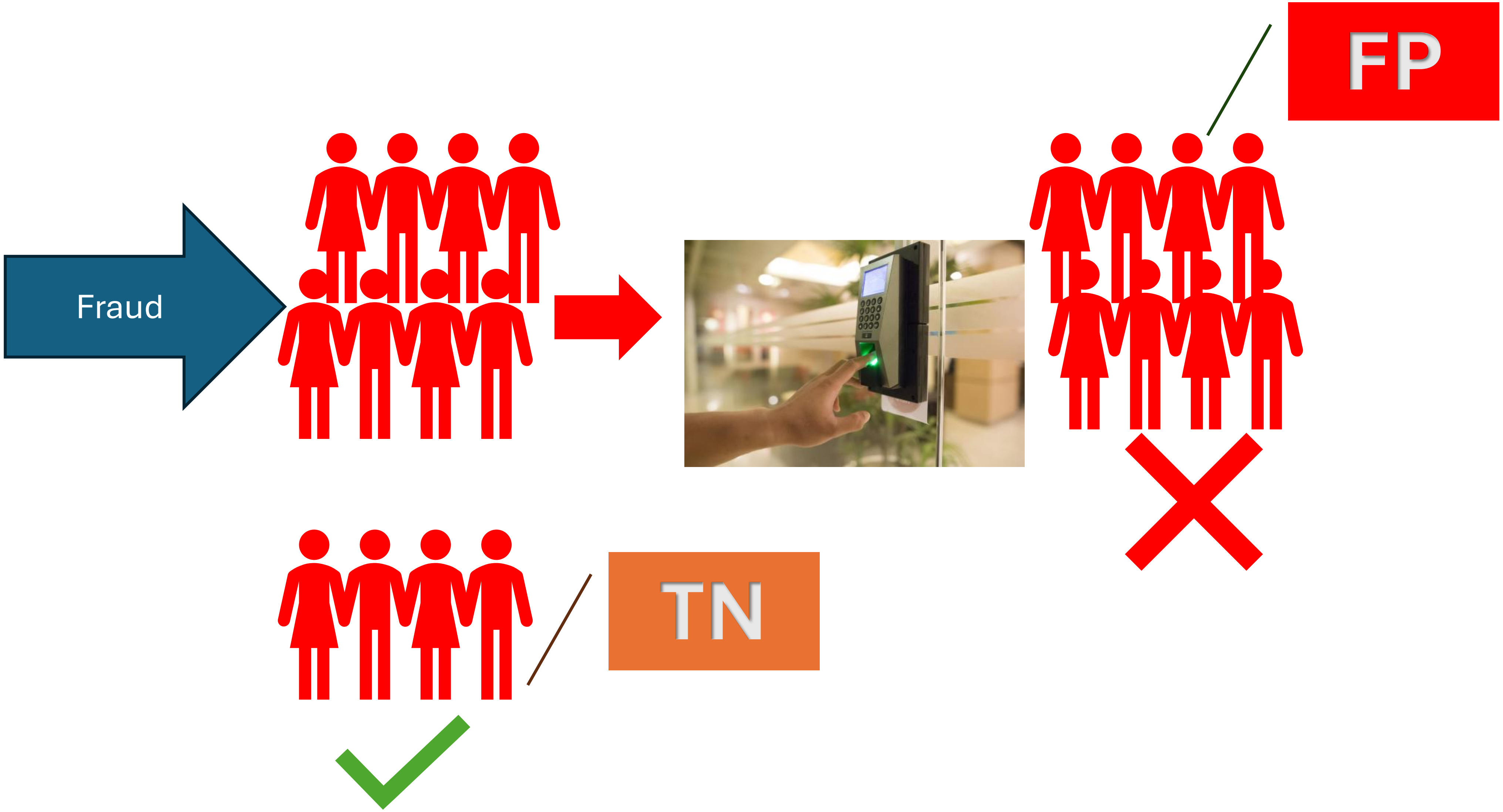
Confusion matrix for binary classification

Actual value	A	TP	FN
	B	FP	TN
		A	B
		Predicted value	

True Positive Vs False Negative



True Positive Vs False Negative



Confusion Matrix

Confusion matrix for binary classification			
Actual value	A	TP	FN
	B	FP	TN
		A	B
		Predicted value	

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

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

Example 1:

	Class A	Class A
Class A	95	5
Class B	50	50
	Predicted	

Given the following values:
•**True Positives (TP) = 95**
•**False Negatives (FN) = 5**
•**False Positives (FP) = 50**
•**True Negatives (TN) = 50**
We can calculate the following metrics:

1. **Precision**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{95}{95 + 50} = \frac{95}{145} \approx 0.655$$

2. **Recall**

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{95}{95 + 5} = \frac{95}{100} = 0.95$$

3. **F1 score:** The F1 score is the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \times \frac{0.655 \times 0.95}{0.655 + 0.95} \approx 2 \times \frac{0.622}{1.605} \approx 0.775$$

Summary of Results:

- **Precision**: ~0.655
- **Recall**: 0.95
- **F1 Score**: ~0.775
- **Accuracy**: 0.725

These results indicate that the system has high recall, and low precision, and overall accuracy, showing a moderate balance between precision and recall. The F1 score further confirms the model’s weakness and unbalancing both metrics.

Example 1:

	Class A	Class B
Class A	95	5
Class B	10	50
	Predicted	

Given:

•True Positives (TP) = 95
•False Negatives (FN) = 5
•False Positives (FP) = 10
•True Negatives (TN) = 50

1. **Precision**

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{95}{95 + 10} = \frac{95}{105} \approx 0.905$$

2. **Recall**

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{95}{95 + 5} = \frac{95}{100} = 0.95$$

3. **F1 score:** The F1 score is the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.905 \times 0.95}{0.905 + 0.95}$$

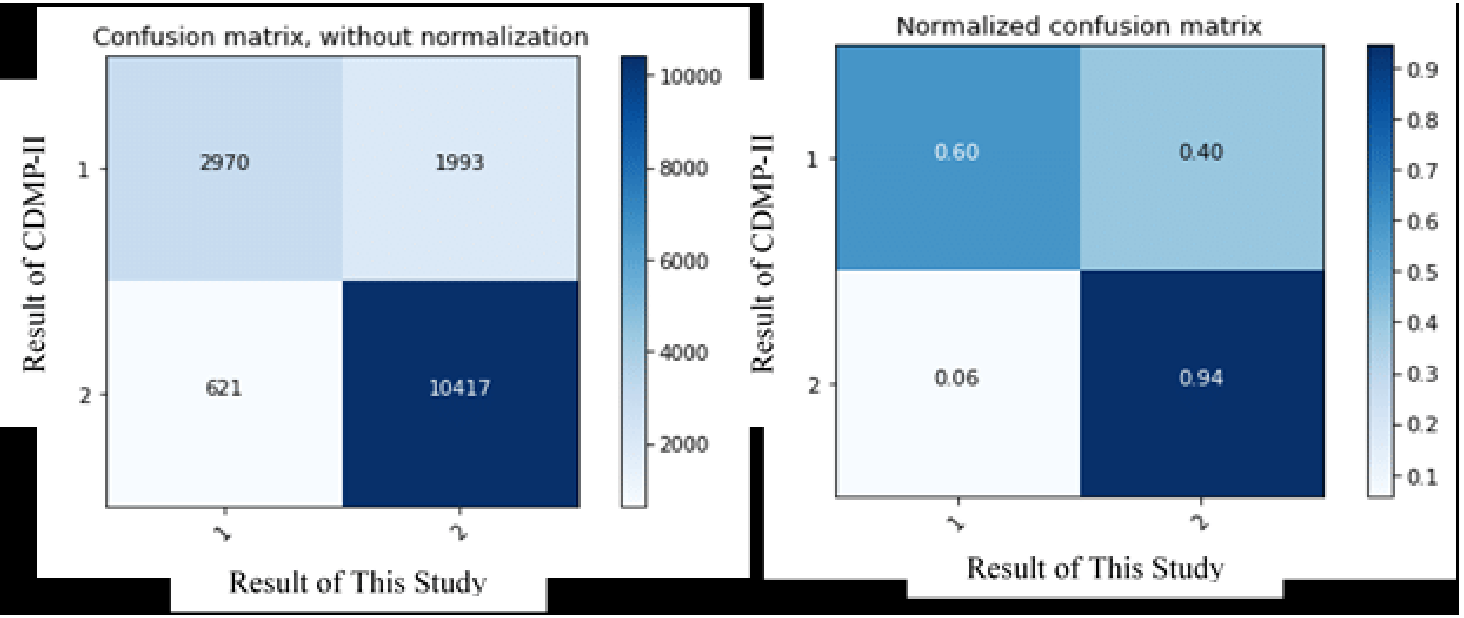
$$\text{F1 Score} \approx 2 \times \frac{0.85975}{1.855} \approx 0.926$$

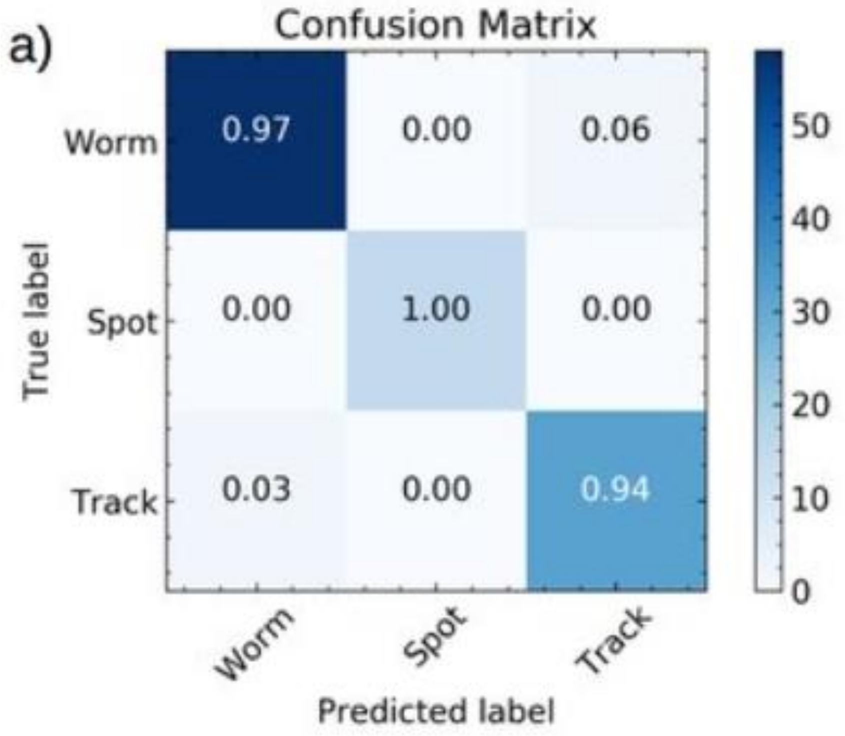
Results:

- **Precision**: ~0.905 (90.5%)
- **Recall**: 0.95 (95%)
- **F1 Score**: ~0.926 (92.6%)
- **Accuracy**: ~0.906 (90.6%)

These results indicate that the system has high recall, precision, and overall accuracy, showing a strong balance between precision and recall. The F1 score further confirms the model's reliability in balancing both metrics.

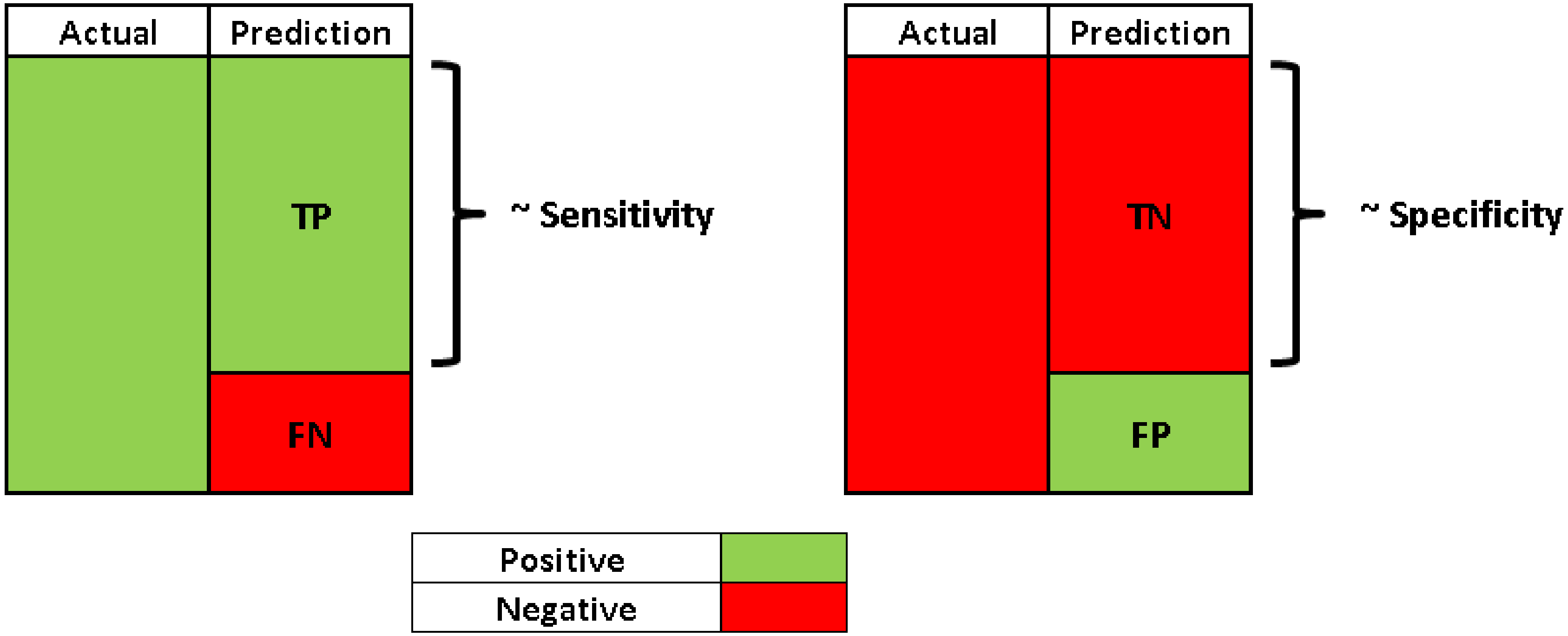
Confusion Matrix





Sensitivity and Specificity

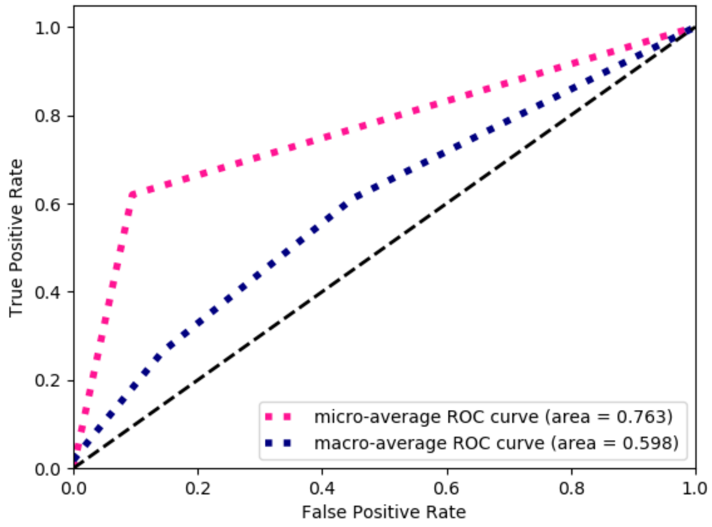
- Sensitivity, also known as the true positive rate (TPR), is the same as recall. Hence, it measures the proportion of positive class that is correctly predicted as positive.
- Specificity is similar to sensitivity but focused on negative class. It measures the proportion of negative class that is correctly predicted as negative.



ROC Curve

- **ROC Curve**: Plots True Positive Rate vs. False Positive Rate.
- **AUC**: Area under the ROC curve; a higher AUC indicates a better model.

- ROC, or Receiver Operating Characteristic, is a graphical representation used to evaluate the performance of a binary classification model. It illustrates the trade-off between sensitivity (true positive rate) and specificity (false positive rate) across different threshold settings. Here's a more detailed breakdown:
- **True Positive Rate (TPR)**:
- Also known as sensitivity or recall.
- It is the proportion of actual positives that are correctly identified by the model.
- **False Positive Rate (FPR)**:
- It is the proportion of actual negatives that are incorrectly identified as positives.
- **Threshold**:
- The probability score at which the model classifies a positive or negative outcome.
- Adjusting the threshold affects the TPR and FPR, creating different points on the ROC curve.
- **ROC Curve**
- The ROC curve plots the TPR against the FPR at various threshold levels.
- The x-axis represents the FPR, while the y-axis represents the TPR.
- Each point on the curve corresponds to a different threshold, showing how the model's predictions change as the threshold is varied.



-

ROC Curve

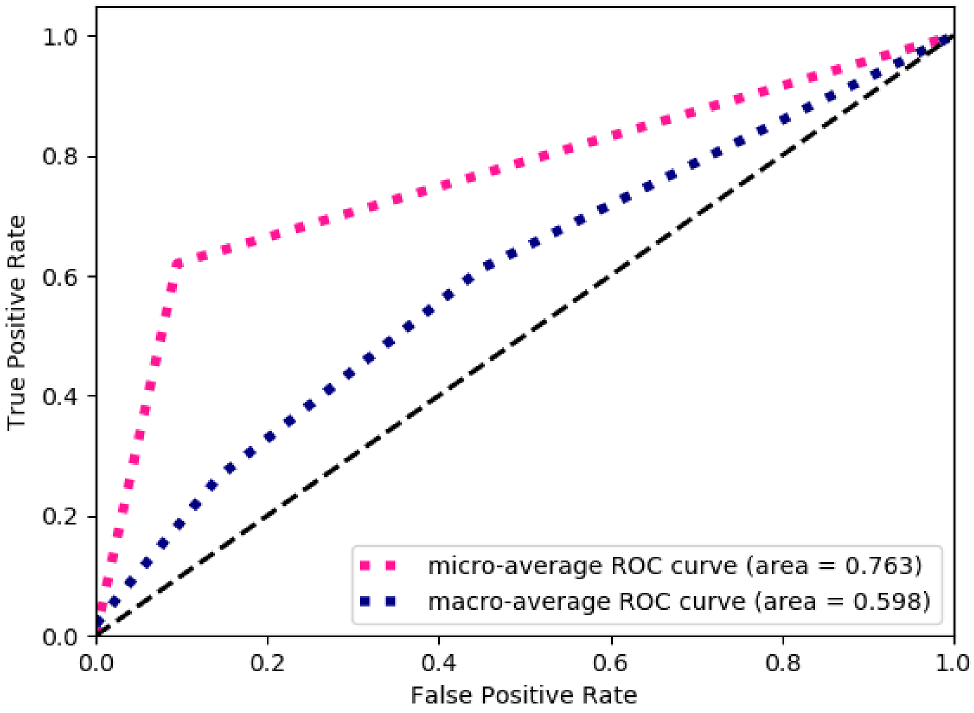
- **Area Under the ROC Curve (AUC)**
- The area under the ROC curve (AUC) quantifies the overall ability of the model to discriminate between positive and negative classes.
- AUC values range from 0 to 1:
- **AUC = 0.5**: The model has no discriminative ability (random guessing).
- **AUC = 1**: The model perfectly distinguishes between classes.
- **AUC < 0.5**: The model is worse than random guessing.

- **Interpretation**
- A model with a higher AUC is generally considered better.
- The ROC curve can also be used to compare multiple models and choose the one that performs best across different thresholds.
- **Example Usage**
- In practice, ROC analysis is widely used in fields like medicine, finance, and machine learning, where it’s crucial to understand the trade-offs between true positives and false positives when making predictions.

Performance Evaluation

Specificity (SEP) = $\frac{\mathrm{TN}}{(\mathrm{TN}+\mathrm{FP})}$

- Sensitivity (SEN) = $\frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}$
- Accuracy (ACC)= $\frac{\mathrm{TN}+\mathrm{TP}}{(\mathrm{TN}+\mathrm{TP}+\mathrm{FN}+\mathrm{FP})}$

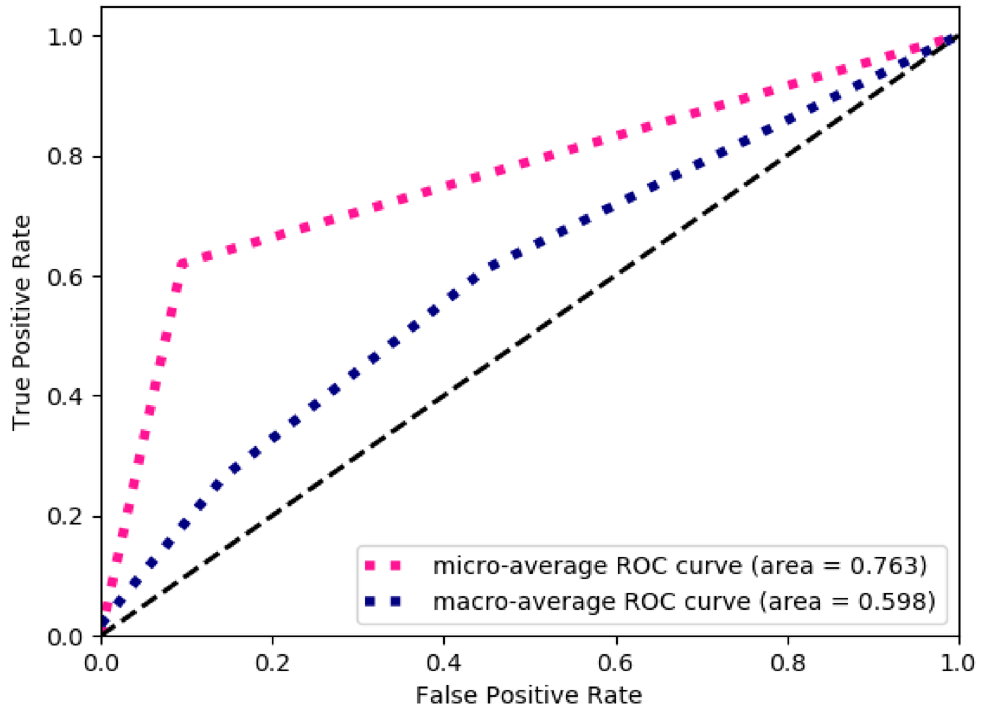


← Object Detection

Performance Evaluation

Evaluation Metrics:

- Receiver Operating Characteristic (ROC) curve
- Area Under the ROC Curve (AUC).



Classification Metrics Comparison

- **Example of Use:** In medical diagnostics, recall might be prioritized to minimize missed diagnoses, while precision might be prioritized in fraud detection to avoid flagging legitimate transactions.

Metric	Best Use Case	Pros	Cons
Accuracy	Balanced datasets with equal class sizes	Simple and intuitive	Misleading for imbalanced datasets
Precision	Cases where false positives are costly	Reduces risk of incorrect positive predictions	Can overlook false negatives
Recall	Cases where false negatives are costly	Emphasizes detecting actual positives	Can result in more false positives
F1-Score	Imbalanced datasets	Balances precision and recall	Can be less interpretable than precision or recall
ROC-AUC	Binary classification, imbalanced datasets	Measures separation between classes	Doesn’t indicate thresholds for specific outcomes

Regression Metrics Comparison

- **Example of Use**: In predicting house prices, MAE might be preferred for a clearer, interpretable measure, while RMSE can highlight how much large errors are impacting the model’s performance.

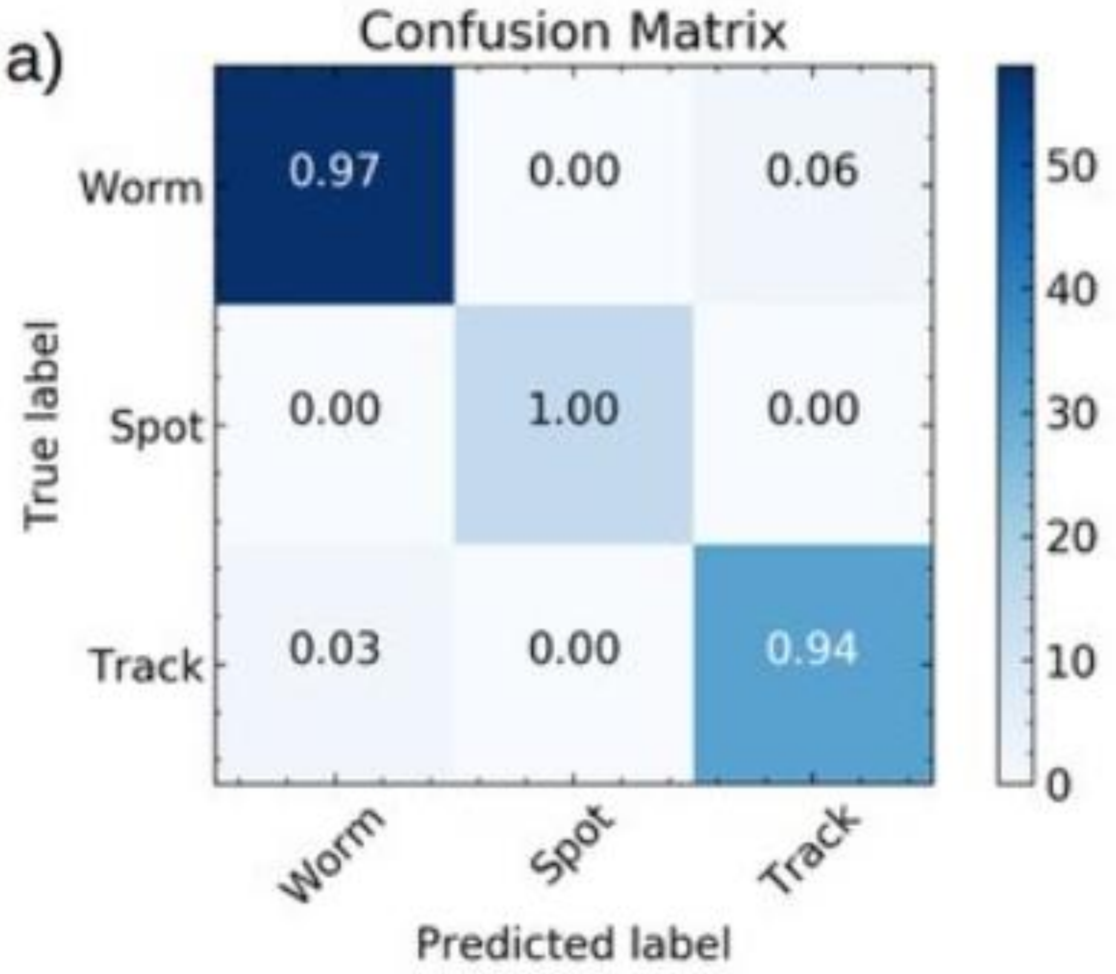
Metric	Best Use Case	Pros	Cons
Mean Absolute Error (MAE)	Real-world interpretation of error	Easy to interpret in original units	May not penalize large errors enough
Mean Squared Error (MSE)	Penalizing larger errors significantly	Highlights large deviations	Squaring errors increases outlier impact
Root Mean Squared Error (RMSE)	Similar to MSE, but in original units	Maintains penalization of large errors	Difficult to interpret if units are abstract
R-squared (R^2)	General variance explanation	Indicates overall model fit	Doesn't indicate the magnitude of residuals

Example

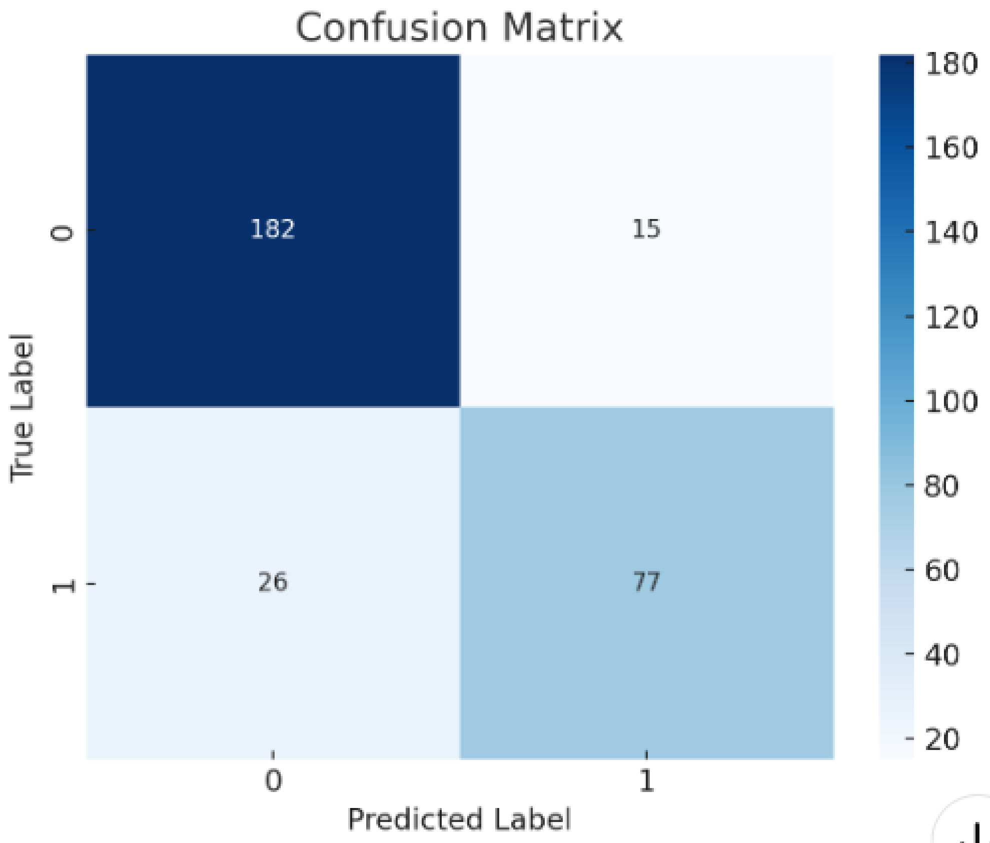
- Let’s go through a practical example of evaluating a classification model using **precision, recall, F1-score**, and the **ROC-AUC**. We’ll use Python with a dataset to classify whether a customer will likely make a purchase, which will help illustrate how these metrics work in real-life applications.
- **Example Overview**

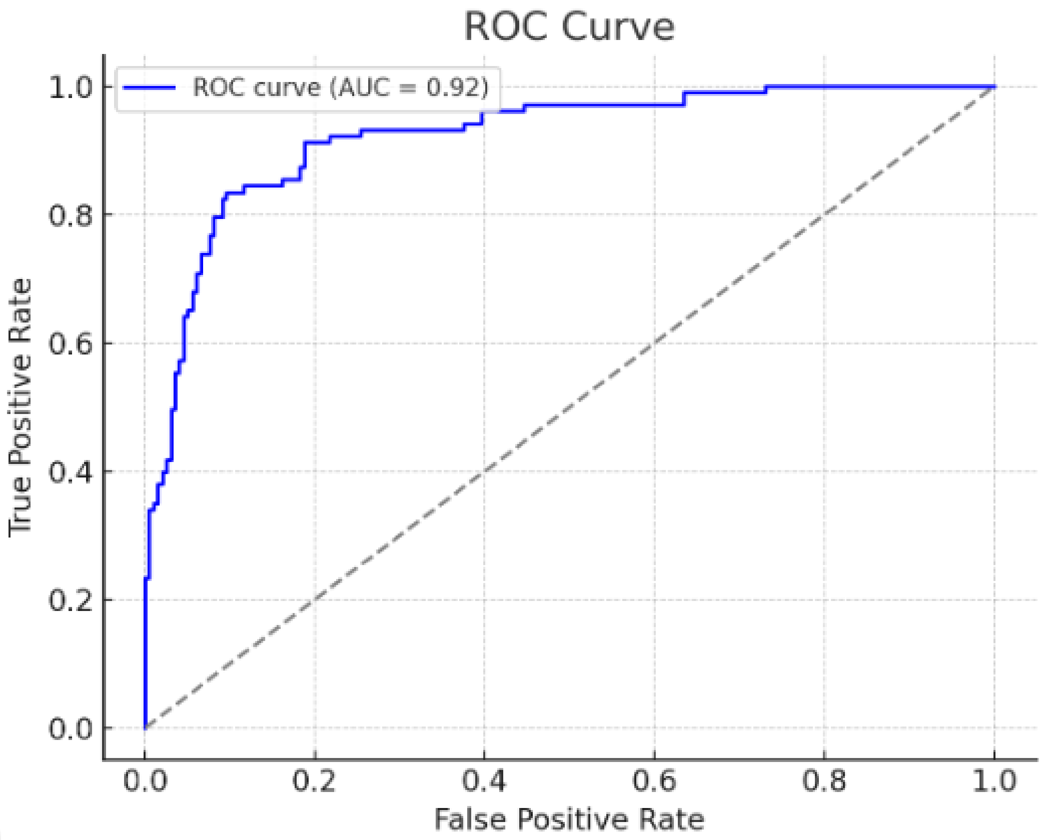
1. **Data**: We’ll use a synthetic dataset (mimicking a "purchase" column) where:
    1. 1 = Made a purchase
    2. 0 = Did not make a purchase
2. **Model**: We’ll train a logistic regression model.
3. **Metrics**: Calculate and interpret the precision, recall, F1-score, and ROC-AUC.

a)


Confusion Matrix
Worm
Spot
Track
True label
0.97
0.00
0.06
0.00
1.00
0.00
0.03
0.00
0.94
Worm
Spot
Track
Predicted label
50
40
30
20
10
0

Here is the practical example, along with the generated metrics and visualizations:
Precision: 0.84
Recall: 0.75
F1-Score: 0.79
ROC-AUC: 0.92





Explanation:

1

Confusion Matrix: The heatmap shows the counts of true positives, true negatives, false positives, and false negatives, which gives a direct overview of the model's performance.

2

Precision: Measures how many of the predicted positive instances (purchases) were correct. Here, precision is 0.84, indicating that 84% of predicted positive cases were correct.

3

Recall: Reflects the model's ability to detect actual positive cases. The recall of 0.75 indicates that 75% of actual purchases were correctly identified.

4

F1-Score: A balanced metric, combining precision and recall. The F1-score of 0.79 shows the model's balance between precision and recall.

5

ROC Curve & AUC: The ROC-AUC of 0.92 shows good model performance, with the curve closer to the top-left corner, indicating a strong balance between true positive and false positive rates.

Classification Metrics Comparison

Metric	Best Use Case	Pros	Cons
Accuracy	Balanced datasets with equal class sizes	Simple and intuitive	Misleading for imbalanced datasets
Precision	Cases where false positives are costly	Reduces risk of incorrect positive predictions	Can overlook false negatives
Recall	Cases where false negatives are costly	Emphasizes detecting actual positives	Can result in more false positives
F1-Score	Imbalanced datasets	Balances precision and recall	Can be less interpretable than precision or recall
ROC-AUC	Binary classification, imbalanced datasets	Measures separation between classes	Doesn’t indicate thresholds for specific outcomes

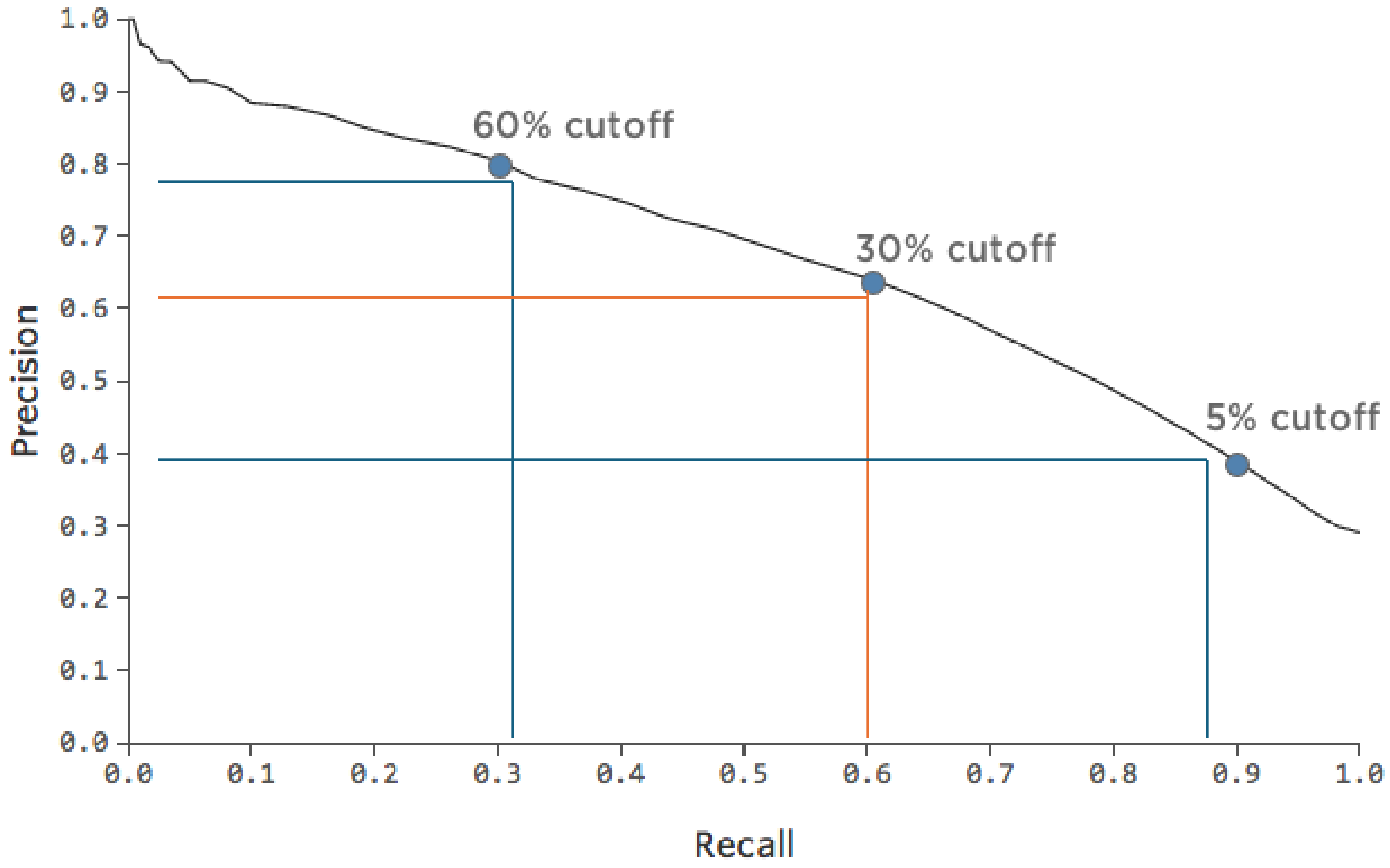
Regression Metrics Comparison

Metric	Best Use Case	Pros	Cons
Mean Absolute Error (MAE)	Real-world interpretation of error	Easy to interpret in original units	May not penalize large errors enough
Mean Squared Error (MSE)	Penalizing larger errors significantly	Highlights large deviations	Squaring errors increases outlier impact
Root Mean Squared Error (RMSE)	Similar to MSE, but in original units	Maintains penalization of large errors	Difficult to interpret if units are abstract
R-squared (R^2)	General variance explanation	Indicates overall model fit	Doesn't indicate the magnitude of residuals

Trading off precision against recall

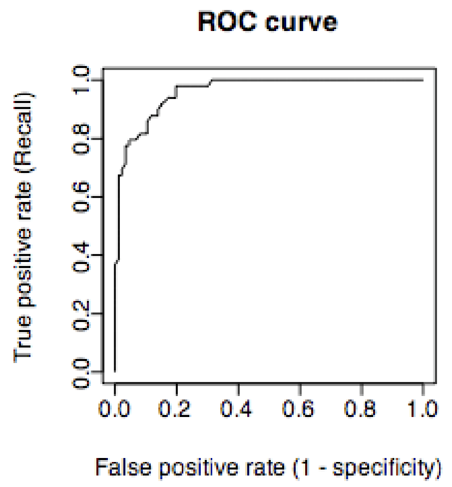
- You can emphasize either precision on the expense of recall or vice versa.
- For example if you are diagnosing people of cancer and you do not want to tell a person that he has cancer while he is not (increase precision). You only diagnose people with high confidence of being ill (may be increase cutoff from 30% to 60%).
  - Precision: 62% —> 80%
    Recall: 60% —> 30%
- Or, if you are interested in not leaving a patient with cancer not diagnosed (increase recall), you may decrease cutoff from 30% to 5%.
  - Precision: 62% —> 40%
    Recall: 60% —> 90%

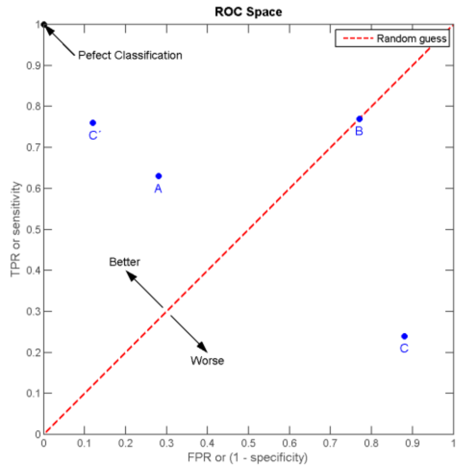
Trading off precision against recall



Receiver Operating Characteristic (ROC) Curve

- The **ROC Curve** is a plot of values of the False Positive Rate (FP) on the x-axis versus the True Positive Rate (TP) on the y-axis for all possible cutoff values from 0% to 100%.
- The higher the ROC curve the better the fit.
- In fact the area under the curve (AUC) can be used for this purpose.
- The closer AUC is to 1 (the maximum value) the better the fit.
- Values close to .5 show that the model's ability to discriminate between success and failure is due to chance.





The Curse of Dimensionality

- As the number of input dimensions gets larger, we will need more data to enable the algorithm to generalize well.
- As the algorithms try to separate data into classes based on the features; therefore as the number of features increases we will need more data points.
- Be careful to understand the data first and not introducing unnecessary features.