EVALUATION METRICS FOR CLASSIFICATION:

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model.

These metrics provide insights into how well the model is performing and help in comparing different models or algorithms.

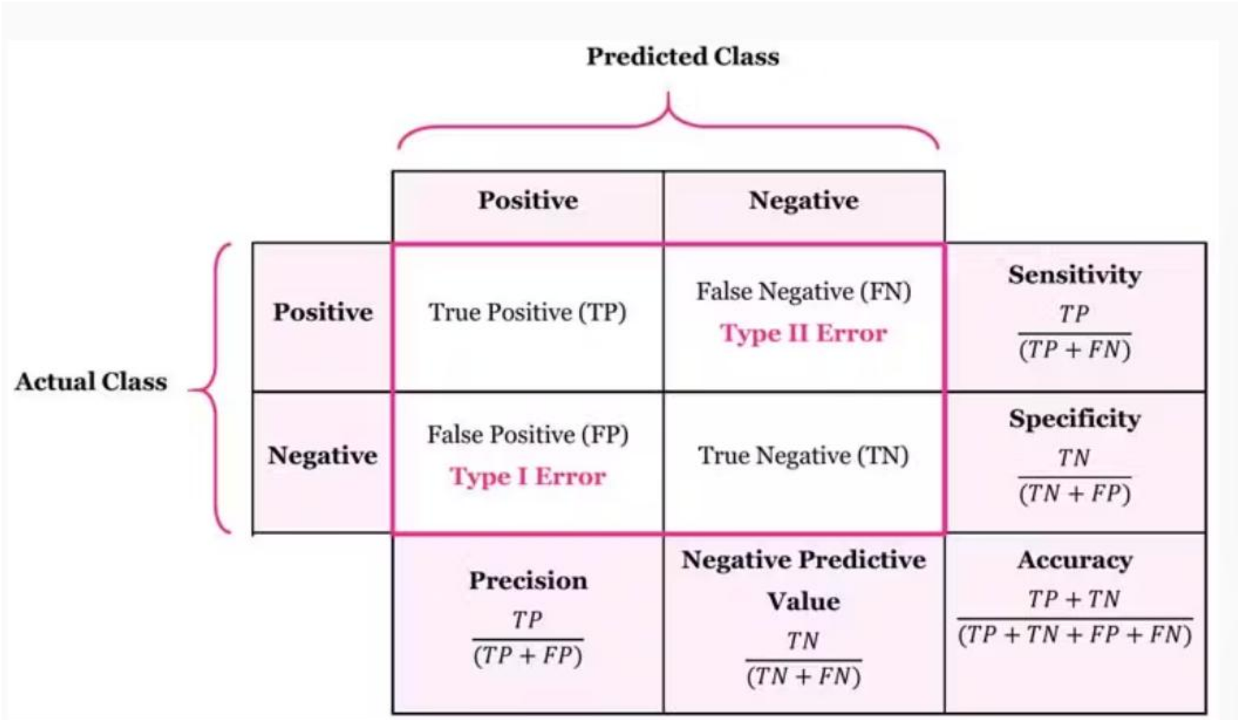
1. Confusion Matrix

- A **Confusion matrix** is an ***N x N matrix*** used for evaluating the **performance of a classification model**, where **N** is the number of ***target classes***.

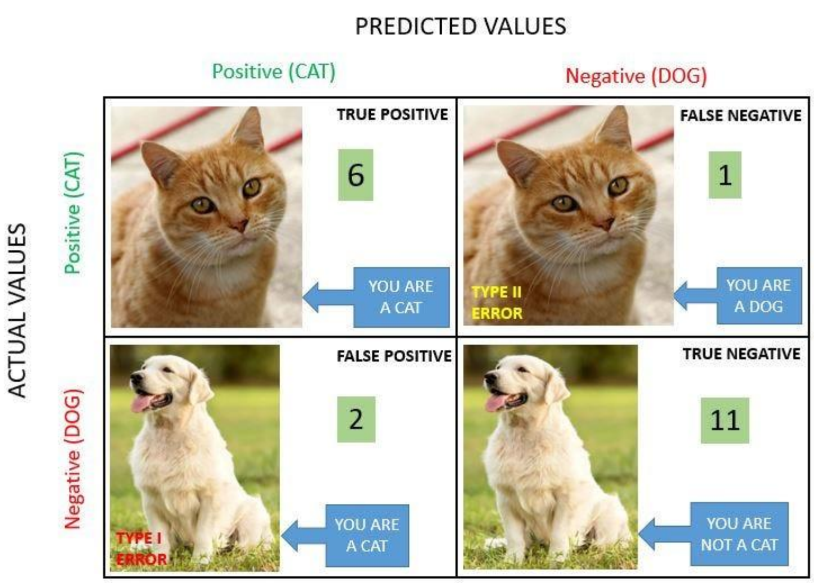
- A table that summarizes predictions vs actual values.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- TP = when the actual value is Positive and predicted is also Positive. (correctly predicted "disease" cases)
- FP = When the actual is negative but prediction is Positive. Also known as the Type 1 error. (healthy people predicted as "diseased" (false alarm))
- FN = When the actual is Positive but the prediction is Negative. Also known as the Type 2 error. (diseased people predicted as "healthy" (missed detection))
- TN = when the actual value is Negative and prediction is also Negative.
(correctly predicted "healthy" cases)



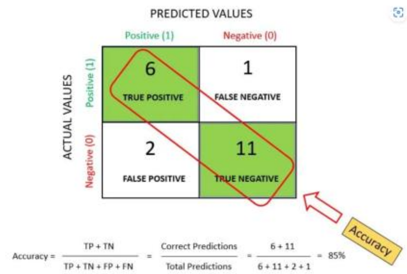
Confusion Matrix



- A good model is one which has *high TP and TN rates*, while *low FP and FN rates*.

2. **Accuracy**

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



- Measures overall correctness. (% of correctly classified samples.)

  90% → Excellent (very strong model, but check class balance!)

  80–90% → Good

  70–80% → Acceptable, but may need improvements

  < 70% → Weak

✓ **Suitable when**:

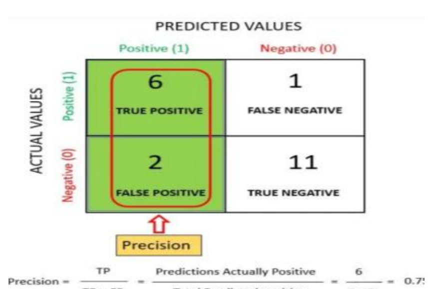
- Classes are **balanced** (roughly equal positive/negative samples).
- The cost of **false positives (FP)** and **false negatives (FN)** is the same.

✗ **Limitations**:

- Misleading for **imbalanced datasets**.
  - Example: If 95% of patients are healthy, predicting "healthy" always gives 95% accuracy → but the model is useless.

3. **Precision (Positive Predictive Value)**

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



- Out of all predicted positives, how many are actually positive?

Useful when **false positives are costly** (e.g., spam detection – don't mark real emails as spam).

Good values:

0.9 → Excellent

0.8–0.9 → Good

Lower → Model struggles to balance false positives/negatives.

✅ **Suitable when**:

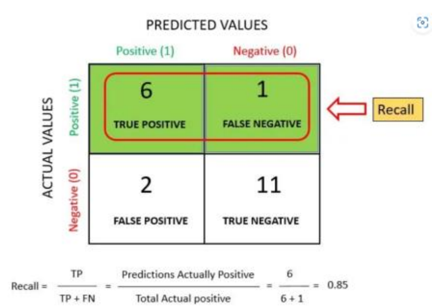
- **False positives are costly**.
- Example:
  - Spam detection → wrongly marking a real email as spam is bad.
  - Fraud detection → flagging an innocent transaction is costly.

❌ **Limitations**:

- Ignores false negatives (doesn't care about missing actual positives).
- Can be high even if the model misses many true positives.

4. **Recall (Sensitivity / True Positive Rate)**

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



- Out of all actual positives, how many did we catch?
- Useful when **false negatives are costly** (e.g., disease detection – don't miss actual patients).

  Good values:

  0.95 → Excellent (very few false negatives)

0.85–0.95 → Good

< 0.8 → Concerning

✅ **Suitable when**:

- **False negatives are costly**.
- Example:
  - Medical tests → missing a cancer patient is dangerous.
  - Security → failing to detect an intruder is critical.

❌ **Limitations**:

- Ignores false positives (predicts almost everything as positive to increase recall).
- Can lead to too many false alarms.

5. **F1-Score**

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

- Harmonic mean of Precision & Recall.
- Good when dataset is **imbalanced**.

Good values:

0.9 → Excellent

0.8–0.9 → Good

Lower → Model struggles to balance false positives/negatives.

✅ **Suitable when**:

- Dataset is **imbalanced**.

- You want a **balance** between Precision and Recall.
- Example: Search engines, recommendation systems.

❌ **Limitations**:

- Doesn't account for **true negatives (TN)**.
- May not fully reflect performance in cases where TN is important (like anomaly detection).

6. Specificity (True Negative Rate)

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- Out of all actual negatives, how many did we correctly identify?
- Useful in **medical screening** (catching healthy people correctly).

✅ **Suitable when**:

- Important to **correctly identify negatives**.
- Example:
  - In medical screening, we want to avoid giving healthy people a false alarm.
  - In malware detection, we don't want to block safe files.

❌ **Limitations**:

- Ignores positives (may miss many true positives).
- Needs to be considered together with recall.

7. **ROC Curve (Receiver Operating Characteristic)**

- Plots True Positive Rate (Recall) **vs** False Positive Rate (FPR = FP / (FP + TN)).
- Shows the trade-off between sensitivity and specificity.

8. **AUC (Area Under ROC Curve)**

- A single value summarizing the ROC curve.
- Ranges between 0 and 1.
- Closer to 1 = better model.

Scale:

- > 0.95 → Outstanding
- 0.90–0.95 → Excellent
- 0.80–0.90 → Good
- 0.70–0.80 → Fair
- < 0.7 → Poor

✅ **Suitable when**:

- Classes are **balanced** or when class distribution is not important.
- Good for **comparing models**.

❌ **Limitations**:

- In **highly imbalanced datasets**, ROC can be overly optimistic.
- PR (Precision-Recall) curve is better when positives are rare.

9. **Log Loss (Cross-Entropy Loss)**

- Measures how far predicted probabilities are from actual class labels.
- Penalizes confident but wrong predictions heavily.
- Lower log loss = better model.

✅ **Suitable when**:

- You care about the **confidence** of predictions.
- Example: Predicting probability of disease → probability 0.95 is more useful than just “Yes”.

❌ **Limitations**:

- Harder to interpret compared to Accuracy/Precision/Recall.
- Very sensitive to incorrect high-confidence predictions.

- **Accuracy → general performance**
- **Precision → when false positives matter**
- **Recall → when false negatives matter**
- **F1-Score → balance between precision & recall**
- **ROC & AUC → probability-based evaluation**
- **Log Loss → probability calibration**

Metric	Best Used When...	Limitation
Accuracy	Balanced classes, equal cost of errors	Misleading on imbalanced data
Precision	FP is costly (spam, fraud)	Ignores FN
Recall	FN is costly (disease detection)	Ignores FP
F1-Score	Need balance between Precision & Recall	Ignores TN
Specificity	Avoiding false alarms is critical	Ignores positives
ROC-AUC	Balanced classes, model comparison	Overly optimistic on imbalanced data
Log Loss	Need probability calibration	Sensitive to wrong confident predictions

- Medical diagnosis → Recall (Sensitivity) is priority.
- Fraud/Spam detection → Precision is priority.
- Balanced tasks → F1 & AUC-ROC are best indicators.

Accuracy Alone is Misleading

- Suppose you build a model to detect a rare disease that affects **1 in 1000 people**.
- A "dumb" model that **always predicts healthy** will be **99.9% accurate** — but it's **useless**, because it never finds the disease.

Different Problems Have Different Priorities

- **Medical Diagnosis** → Missing a sick patient (False Negative) is worse → we care about **Recall**.
- **Spam Detection** → Flagging important emails as spam (False Positive) is worse → we care about **Precision**.
- **Search Engines / Recommendations** → We care about a balance between precision & recall → use **F1-score, AUC**.

Why Evaluation Metrics

- **To Handle Class Imbalance**
  - If one class dominates (e.g., fraud = 1%, non-fraud = 99%), then a naive model can look "accurate" without actually solving the problem.

    ☞ Metrics like **ROC-AUC, Precision-Recall AUC, F1-score** help evaluate correctly.
- **Understand Model Behavior Beyond "Right or Wrong"**
  - Metrics let us ask:

    - Is my model **over-predicting positives** (high FP)?
    - Is it **missing too many actual positives** (high FN)?
    - Is it consistent across different thresholds?

For Model Comparison & Selection

- When we train multiple models (say Logistic Regression, Decision Tree, Random Forest), we need **metrics** to compare and pick the best one for the task.