FOR CLASSIFICATION

1. Confusion Matrix

A **Confusion Matrix** is a table used to evaluate the performance of a classification model. It compares the actual and predicted values and provides insights into errors.

Confusion Matrix Table (for Binary Classification)

Actual / Predicted	Predicted: Positive (1)	Predicted: Negative (0)
Actual: Positive (1)	**True Positive (TP)**	**False Negative (FN)**
Actual: Negative (0)	**False Positive (FP)**	**True Negative (TN)**

Explanation of Terms:

- **True Positive (TP)** → Model correctly predicts a positive class (e.g., detecting a disease correctly).
- **False Positive (FP) (Type I Error)** → Model incorrectly predicts a positive class (e.g., diagnosing a healthy person as sick).
- **False Negative (FN) (Type II Error)** → Model incorrectly predicts a negative class (e.g., failing to detect a disease in a sick patient).
- **True Negative (TN)** → Model correctly predicts a negative class (e.g., correctly identifying a healthy person).

2. Evaluation Metrics for Classification

Once we have a confusion matrix, we can derive important evaluation metrics:

1. Accuracy

Accuracy = (TP+TN) / (TP +FP+FN+TN)

✅ **When to Use:** When the dataset is balanced (equal number of positive and negative classes).
❌ **When Not to Use:** If the dataset is imbalanced (e.g., 95% healthy, 5% sick), accuracy can be misleading.

2. Precision (Positive Predictive Value)

Precision = TP / (FP + TP)

✅ **When to Use:** When **false positives are costly** (e.g., spam detection, where classifying a legitimate email as spam is bad).

3. Recall (Sensitivity or True Positive Rate)

Recall = TP / (FN + TP)

✅ **When to Use:** When **false negatives are costly** (e.g., medical diagnosis, where missing a disease could be dangerous).

4. F1-Score (Harmonic Mean of Precision & Recall)

F1-Score = (2 × Precision × Recall) / (Precision+Recall)

✅ **When to Use:** When there is an **imbalance** between positive and negative classes.
❌ **When Not to Use:** If precision or recall alone is more important.

5. Specificity (True Negative Rate)

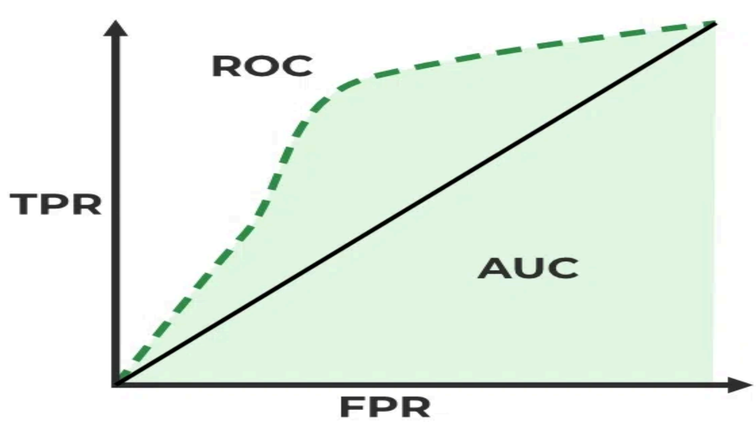
Formula:

Specificity = TN / (TN+FP)

✅ **When to Use:** When detecting **negative cases is crucial** (e.g., fraud detection, where falsely accusing someone is problematic).

6. ROC Curve (Receiver Operating Characteristic) & AUC (Area Under Curve)

- **ROC Curve** → Plots **True Positive Rate (TPR) vs. False Positive Rate (FPR)** at different thresholds.
- **AUC (Area Under Curve)** → Measures how well the model separates classes.
  ✅ **When to Use:** Comparing multiple models, choosing thresholds.



Example Code: Confusion Matrix & Evaluation Metrics

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
precision_score, recall_score, f1_score

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.datasets import make_classification

# Generate sample classification dataset

X, y = make_classification(n_samples=1000, n_features=10, random_state=42)

# Split into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Train a model

model = RandomForestClassifier()

model.fit(X_train, y_train)

# Make predictions

y_pred = model.predict(X_test)

# Compute confusion matrix

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:")

print(cm)

# Compute evaluation metrics

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)
```

```python
f1 = f1_score(y_test, y_pred)

print("\nEvaluation Metrics:")

print(f"Accuracy: {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1-Score: {f1:.2f}")

# Detailed classification report

print("\nClassification Report:")

print(classification_report(y_test, y_pred))
```

When to Use Each Metric?

Metric	When to Use?
Accuracy	When classes are balanced.
Precision	When false positives are costly (e.g., spam filtering, fraud detection).
Recall	When false negatives are costly (e.g., medical diagnoses, security alarms).
F1-Score	When we need a balance between Precision and Recall.
ROC-AUC	When comparing multiple models.

FOR REGRESSION

Metric	When to USE	Pros	Cons
MAE	When you need an easy-to-understand average error.	Simple to interpret, not sensitive to large errors.	Treats all errors equally, which may not always be ideal
MSE	When you want to penalize large errors more than small ones.	Penalizes large errors heavily.	Squaring makes the error larger, making it less interpretable.
RMSE	When you want an interpretable metric in the same unit as the target variable.	More interpretable than MSE.	Still penalizes large errors.
R-sqaure	When you want to measure how well the model explains variance.	Helps compare models, provides a scale-independent measure.	Can be misleading if dataset is too small.