Regression metrics are used to evaluate how well a regression model (predicting continuous values) performs. Here are the **most commonly used regression metrics**, along with formulas and explanations:

? 1. Mean Absolute Error (MAE)

- **Formula**:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} | x_i - x |$$

- **Meaning**: Average of the absolute differences between predicted and actual values.
- **Range**: [0, ∞)
- **Lower is better**.

? 2. Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

- **Meaning**: Penalizes larger errors more than MAE due to squaring.
- **Range**: [0, ∞)
- **Lower is better**.

---

3. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

- **Meaning**: Square root of MSE, in the same units as target variable.
- **Lower is better**.

---

4. **R-squared (R^2) or Coefficient of Determination**

$$R^2 = 1 - \frac{\sum_{i=i}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=i}^{n}(y_i - \bar{y})^2}$$

- **Meaning**: Proportion of variance in the target explained by the model.
- **Range**: $(-\infty, 1]$
- **Closer to 1 is better**. A value of 0 means the model does no better than the mean.

Example in Python:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

y_true = [3.0, -0.5, 2.0, 7.0]
y_pred = [2.5, 0.0, 2.1, 7.8]

mae = mean_absolute_error(y_true, y_pred)
mse = mean_squared_error(y_true, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_true, y_pred)

print(f"MAE: {mae:.2f}")
print(f"MSE: {mse:.2f}")
print(f"RMSE: {rmse:.2f}")
print(f"R²: {r2:.2f}")
```

Classification metrics help evaluate the performance of classification models (binary or multiclass).

1. Accuracy

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

- **Meaning**: Proportion of total correct predictions.
- **Best for**: Balanced datasets.

Example: COVID-19 Test Prediction

You developed a machine learning model to detect if a person has COVID-19 (positive) or not (negative). The actual condition and predicted result are compared for 100 people.

Definitions in This Context:

- **True Positive (TP = 40)**: 40 people actually had COVID, and the model correctly predicted positive.
- **False Negative (FN = 10)**: 10 people had COVID, but the model predicted negative (missed cases).
- **False Positive (FP = 5)**: 5 people didn't have COVID, but the model falsely predicted positive.
- **True Negative (TN = 45)**: 45 people didn't have COVID, and the model correctly predicted negative.

✅ 2. Precision

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

- **Meaning**: Out of all predicted positives, how many are actually positive?
- **Best for**: When **false positives** are costly (e.g., spam detection).

✅ 3. Recall (Sensitivity / True Positive Rate)

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

- **Meaning**: Out of all actual positives, how many were correctly predicted?
- **Best for**: When **false negatives** are costly (e.g., disease detection).

✅ 4. F1-Score

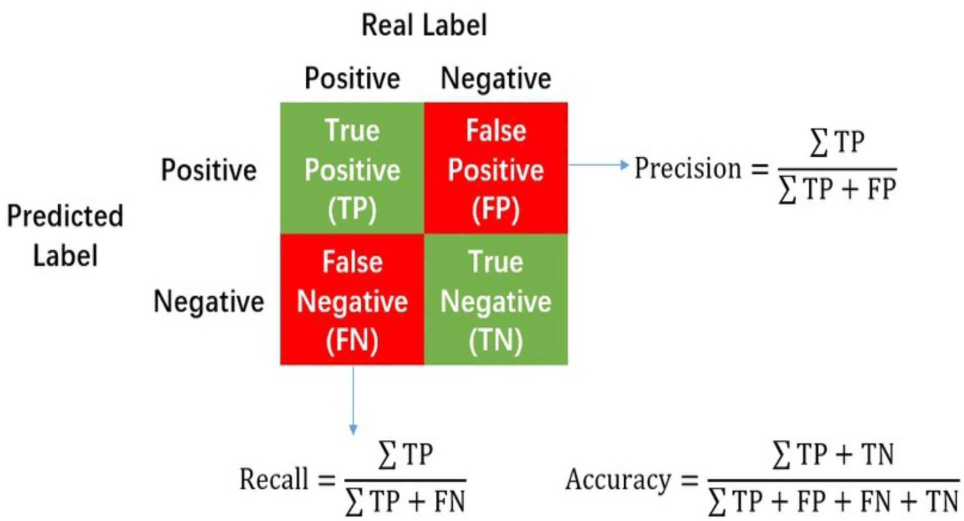
- **Formula**:

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

- **Meaning**: Harmonic mean of precision and recall.
- **Best for**: Imbalanced classes.

✓5. Confusion Matrix

- A matrix showing:
    - **True Positives (TP)**
    - **False Positives (FP)**
    - **True Negatives (TN)**
    - **False Negatives (FN)**



✓6. ROC AUC Score

- **ROC Curve** plots TPR vs. FPR at different thresholds.
- **AUC (Area Under Curve)**: Closer to 1 is better.
- **Best for**: Binary classification performance comparison.

Youtube Link :https://www.youtube.com/watch?v=4jRBRDbJemM&pp=0gcJCfwAo7VqN5tD