**Evaluation Metrics for Classification:**

- **Confusion Matrix:** This table summarizes the performance of a classification model on a test dataset. It shows the number of correct and incorrect predictions for each category.
- **Accuracy:** The overall percentage of correct predictions made by the model.
- **Error Rate:** The percentage of incorrect predictions.
- **Precision:** The proportion of positive predictions that were actually correct.
- **Recall (Sensitivity):** The proportion of actual positive cases that were correctly identified by the model.
- **Specificity:** The proportion of actual negative cases that were correctly identified by the model.

**1. Confusion Matrix:**

Imagine a scenario where you're training a spam filter (classification task). The confusion matrix would be a 2x2 table summarizing how well your filter classifies emails:

| Predicted Label | Actual Spam | Actual Not Spam |
|---|---|---|
| **Spam** (TP) | True Positives (TP): Emails correctly classified as spam. | False Positives (FP): Emails incorrectly classified as spam (Type I error). |
| **Not Spam** (TN) | False Negatives (FN): Spam emails incorrectly classified as not spam (Type II error). | True Negatives (TN): Emails correctly classified as not spam. |

**Example:**

| Predicted Label | Actual Spam | Actual Not Spam |
|---|---|---|
| **Spam** (TP) | 80 | 5 (FP) |
| **Not Spam** (TN) | 10 (FN) | 995 (TN) |

This example shows the filter correctly classified 80 spam emails (TP) but mistakenly flagged 5 non-spam emails as spam (FP). It also missed 10 spam emails (FN) but correctly identified 995 legitimate emails (TN).

**FN : Type-1 Error**

**FP : Type -2 Error**

### 2. Accuracy:

- **Formula:** Accuracy = (TP + TN) / (Total Number of Emails)
- **Interpretation:** Overall percentage of correctly classified emails.

**Example Accuracy:**

Using the example confusion matrix, Accuracy = (80 + 995) / (1000) = 0.875 or 87.5%

### 3. Error Rate:

- **Formula:** Error Rate = 1 - Accuracy
- **Interpretation:** Percentage of incorrectly classified emails.

**Example Error Rate:**

Error Rate = 1 - 0.875 = 0.125 or 12.5%

### 4. Precision:

- **Formula:** Precision = TP / (TP + FP)
- **Interpretation:** Proportion of predicted spam emails that were actually spam.

**Example Precision:**

Precision = 80 / (80 + 5) = 0.941 or 94.1%

### 5. Recall (Sensitivity):

- **Formula:** Recall = TP / (TP + FN)
- **Interpretation:** Proportion of actual spam emails that were correctly classified as spam.

**Example Recall:**

Recall = 80 / (80 + 10) = 0.889 or 88.9%

### 6. Specificity:

- **Formula:** Specificity = TN / (TN + FP)
- **Interpretation:** Proportion of actual non-spam emails that were correctly classified as not spam.

**Example Specificity:**

Specificity = 995 / (995 + 5) = 0.99 or 99%

**Choosing the Right Metric:**

The most appropriate metric depends on the specific application. Here's a general guideline:

- **Accuracy:** Good overall measure, but can be misleading in imbalanced datasets (e.g., very few spam emails).
- **Precision:** Useful when the cost of false positives is high (e.g., medical diagnosis).
- **Recall:** Important when missing true positives is critical (e.g., fraud detection).
- **Specificity:** Important when incorrectly classifying negatives is costly (e.g., spam filter blocking important emails).

By considering these metrics together, you can gain a comprehensive understanding of your classification model's performance and identify areas for improvement.

**EXAMPLE 2**

Imagine you're training a machine learning model to classify images as containing either cats or dogs. Here's how the confusion matrix and evaluation metrics would work:

**SConfusion Matrix:**

| **Predicted Label** | **Actual Cat** | **Actual Dog** |
|---|---|---|
| **Cat** (TP) | True Positives (TP): Images correctly classified as cats. | False Positives (FP): Images of dogs incorrectly classified as cats. |
| **Dog** (TN) | False Negatives (FN): Cat images incorrectly classified as dogs. | True Negatives (TN): Images of dogs correctly classified as dogs. |

drive_spreadsheetExport to Sheets

**Example:**

| **Predicted Label** | **Actual Cat** | **Actual Dog** |
|---|---|---|
| **Cat** (TP) | 200 | 10 (FP) |
| **Dog** (TN) | 5 (FN) | 285 (TN) |

drive_spreadsheetExport to Sheets

**Evaluation Metrics:**

1. **Accuracy:**

- **Formula:** Accuracy = (TP + TN) / (Total Number of Images)
- **Interpretation:** Overall percentage of correctly classified images.

**Example Accuracy:**

Accuracy = (200 + 285) / (500) = 0.97 or 97%

2. **Error Rate:**

- **Formula:** Error Rate = 1 - Accuracy
- **Interpretation:** Percentage of incorrectly classified images.

**Example Error Rate:**

Error Rate = 1 - 0.97 = 0.03 or 3%

3. **Precision (for Cats):**

- **Formula:** Precision = TP / (TP + FP)
- **Interpretation:** Proportion of predicted cat images that were actually cats.

**Example Precision (Cats):**

Precision (Cats) = 200 / (200 + 10) = 0.952 or 95.2%

4. **Recall (Sensitivity) (for Cats):**

- **Formula:** Recall = TP / (TP + FN)
- **Interpretation:** Proportion of actual cat images that were correctly classified as cats.

**Example Recall (Cats):**

Recall (Cats) = 200 / (200 + 5) = 0.976 or 97.6%

5. **Specificity (for Dogs):**

- **Formula:** Specificity = TN / (TN + FP)
- **Interpretation:** Proportion of actual dog images that were correctly classified as dogs.

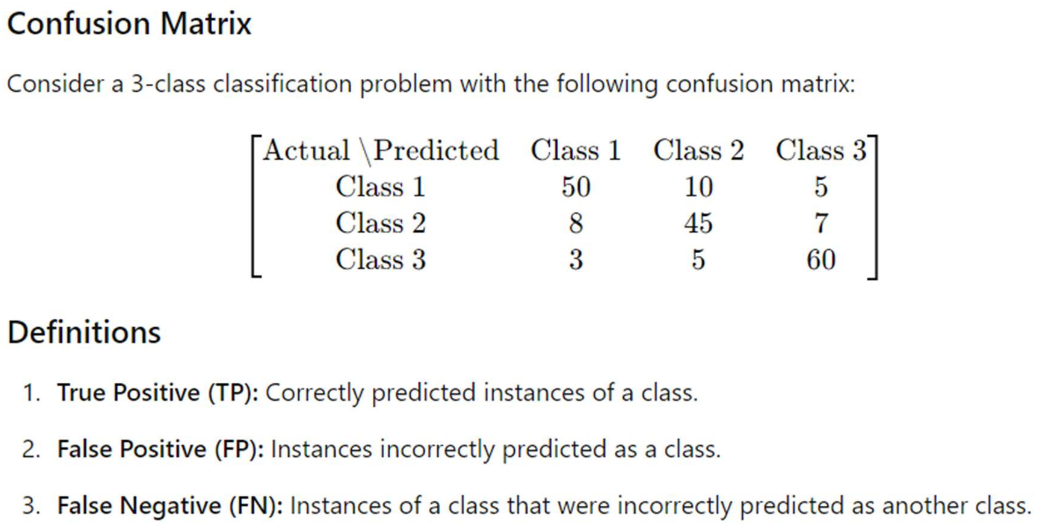
**Example Specificity (Dogs):**

Specificity (Dogs) = 285 / (285 + 10) = 0.966 or 96.6%

**Analysis:**

This example shows a high overall accuracy (97%), indicating the model performs well in classifying most images correctly. However, there are some misclassifications:

- **False Positives (10):** The model mistakenly classified 10 dog images as cats. This could be an issue if it's crucial to avoid classifying dogs as cats (e.g., an adoption website). Here, precision for cats (95.2%) tells us that most predicted cat images are indeed cats, but there's still room for improvement.
- **False Negatives (5):** The model missed 5 cat images, classifying them as dogs. Recall for cats (97.6%) is high, but these missed images could be important depending on the application.



## Confusion Matrix

Consider a 3-class classification problem with the following confusion matrix:

$$\begin{bmatrix} \text{Actual} \setminus \text{Predicted} & \text{Class 1} & \text{Class 2} & \text{Class 3} \\ \text{Class 1} & 50 & 10 & 5 \\ \text{Class 2} & 8 & 45 & 7 \\ \text{Class 3} & 3 & 5 & 60 \end{bmatrix}$$

## Definitions

1. **True Positive (TP):** Correctly predicted instances of a class.
2. **False Positive (FP):** Instances incorrectly predicted as a class.
3. **False Negative (FN):** Instances of a class that were incorrectly predicted as another class.

**Class 1**

- TP1 = 50
- FP1 = 10 + 5 = 15
- FN1 = 8 + 3 = 11

**Precision (Class 1):**

$$\text{Precision}_1 = \frac{TP1}{TP1 + FP1} = \frac{50}{50 + 15} = \frac{50}{65} \approx 0.769$$

**Recall (Class 1):**

$$\text{Recall}_1 = \frac{TP1}{TP1 + FN1} = \frac{50}{50 + 11} = \frac{50}{61} \approx 0.820$$

**F1 Score (Class 1):**

$$F1_1 = 2 \times \frac{\text{Precision}_1 \times \text{Recall}_1}{\text{Precision}_1 + \text{Recall}_1} = 2 \times \frac{0.769 \times 0.820}{0.769 + 0.820} \approx 0.794$$

**Class 2**

- TP2 = 45
- FP2 = 8 + 7 = 15
- FN2 = 10 + 5 = 15

**Precision (Class 2):**

$$\text{Precision}_2 = \frac{TP2}{TP2 + FP2} = \frac{45}{45 + 15} = \frac{45}{60} = 0.75$$

**Recall (Class 2):**

$$\text{Recall}_2 = \frac{TP2}{TP2 + FN2} = \frac{45}{45 + 15} = \frac{45}{60} = 0.75$$

**F1 Score (Class 2):**

$$F1_2 = 2 \times \frac{\text{Precision}_2 \times \text{Recall}_2}{\text{Precision}_2 + \text{Recall}_2} = 2 \times \frac{0.75 \times 0.75}{0.75 + 0.75} = 0.75$$

**Class 3**

- TP3 = 60
- FP3 = 5 + 7 = 12
- FN3 = 5 + 7 = 12

**Precision (Class 3):**

$$\text{Precision}_3 = \frac{TP3}{TP3 + FP3} = \frac{60}{60 + 12} = \frac{60}{72} \approx 0.833$$

**Recall (Class 3):**

$$\text{Recall}_3 = \frac{TP3}{TP3 + FN3} = \frac{60}{60 + 12} = \frac{60}{72} \approx 0.833$$

**F1 Score (Class 3):**

$$F1_3 = 2 \times \frac{\text{Precision}_3 \times \text{Recall}_3}{\text{Precision}_3 + \text{Recall}_3} = 2 \times \frac{0.833 \times 0.833}{0.833 + 0.833} = 0.833$$

## Averaging Methods

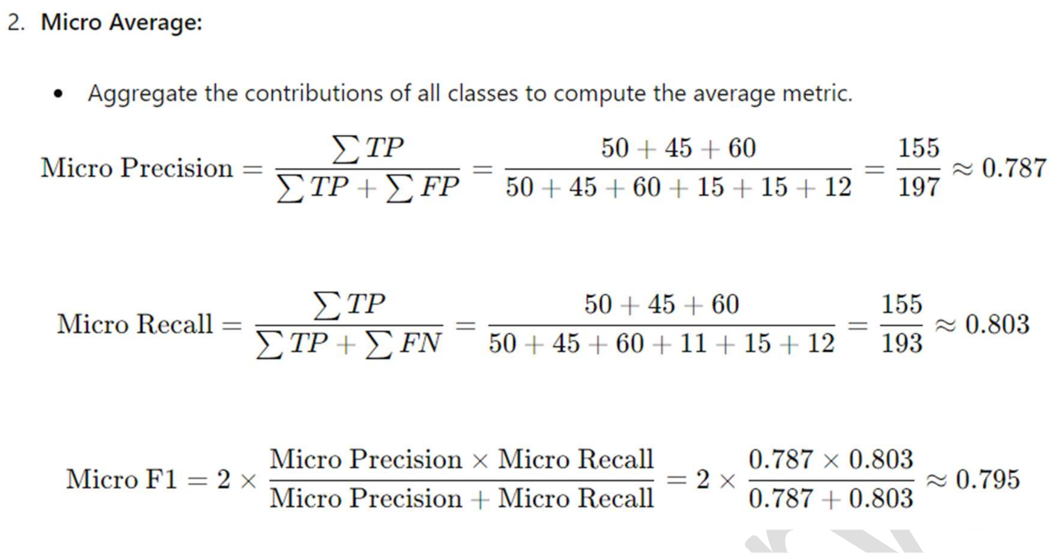
1. **Macro Average:**

   - Calculate the metric independently for each class and then take the average.

$$\text{Macro Precision} = \frac{\text{Precision}_1 + \text{Precision}_2 + \text{Precision}_3}{3} \approx \frac{0.769 + 0.75 + 0.833}{3} \approx 0.784$$

$$\text{Macro Recall} = \frac{\text{Recall}_1 + \text{Recall}_2 + \text{Recall}_3}{3} \approx \frac{0.820 + 0.75 + 0.833}{3} \approx 0.801$$

$$\text{Macro F1} = \frac{F1_1 + F1_2 + F1_3}{3} \approx \frac{0.794 + 0.75 + 0.833}{3} \approx 0.792$$



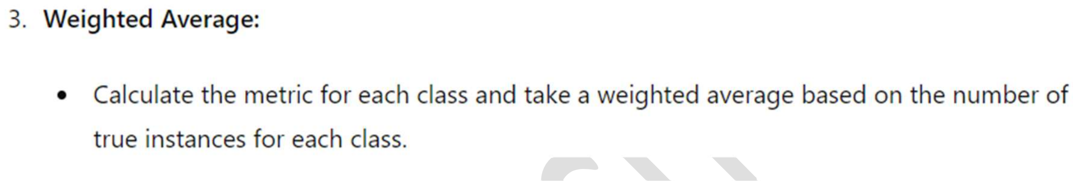
2. **Micro Average:**

- Aggregate the contributions of all classes to compute the average metric.

$$\text{Micro Precision} = \frac{\sum TP}{\sum TP + \sum FP} = \frac{50+45+60}{50+45+60+15+15+12} = \frac{155}{197} \approx 0.787$$

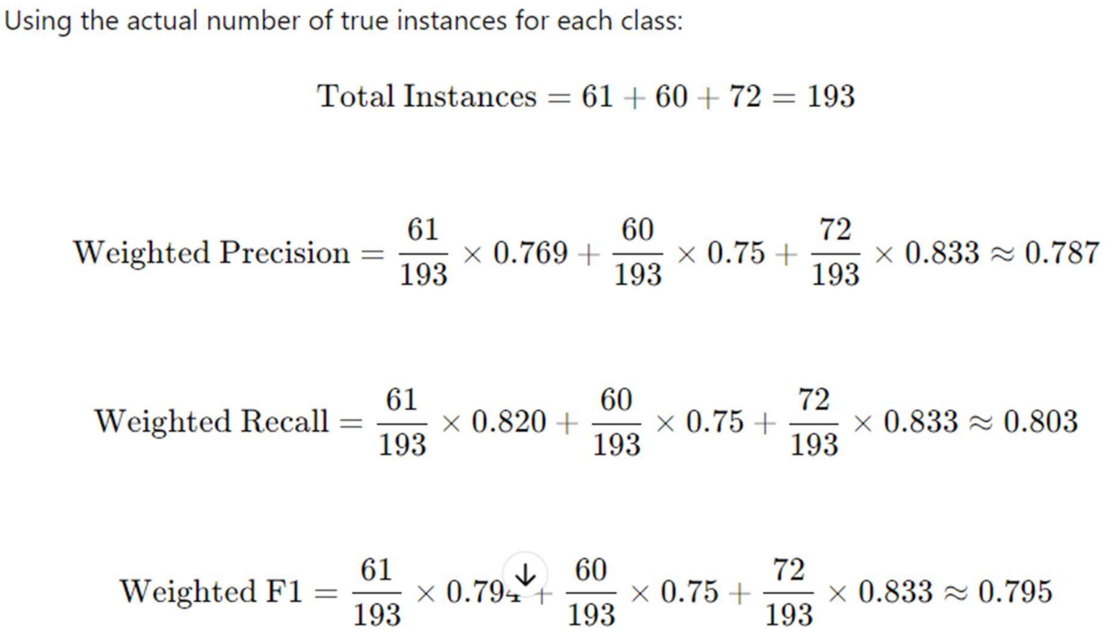
$$\text{Micro Recall} = \frac{\sum TP}{\sum TP + \sum FN} = \frac{50+45+60}{50+45+60+11+15+12} = \frac{155}{193} \approx 0.803$$

$$\text{Micro F1} = 2 \times \frac{\text{Micro Precision} \times \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}} = 2 \times \frac{0.787 \times 0.803}{0.787 + 0.803} \approx 0.795$$



3. **Weighted Average:**

- Calculate the metric for each class and take a weighted average based on the number of true instances for each class.



Using the actual number of true instances for each class:

$$\text{Total Instances} = 61 + 60 + 72 = 193$$

$$\text{Weighted Precision} = \frac{61}{193} \times 0.769 + \frac{60}{193} \times 0.75 + \frac{72}{193} \times 0.833 \approx 0.787$$

$$\text{Weighted Recall} = \frac{61}{193} \times 0.820 + \frac{60}{193} \times 0.75 + \frac{72}{193} \times 0.833 \approx 0.803$$

$$\text{Weighted F1} = \frac{61}{193} \times 0.794 + \frac{60}{193} \times 0.75 + \frac{72}{193} \times 0.833 \approx 0.795$$

## Summary of Averages

- **Macro Average:**
  - Precision: 0.784
  - Recall: 0.801
  - F1 Score: 0.792
- **Micro Average:**
  - Precision: 0.787

    - Recall: 0.803
    - F1 Score: 0.795
- **Weighted Average:**
    - Precision: 0.787
    - Recall: 0.803
    - F1 Score: 0.795