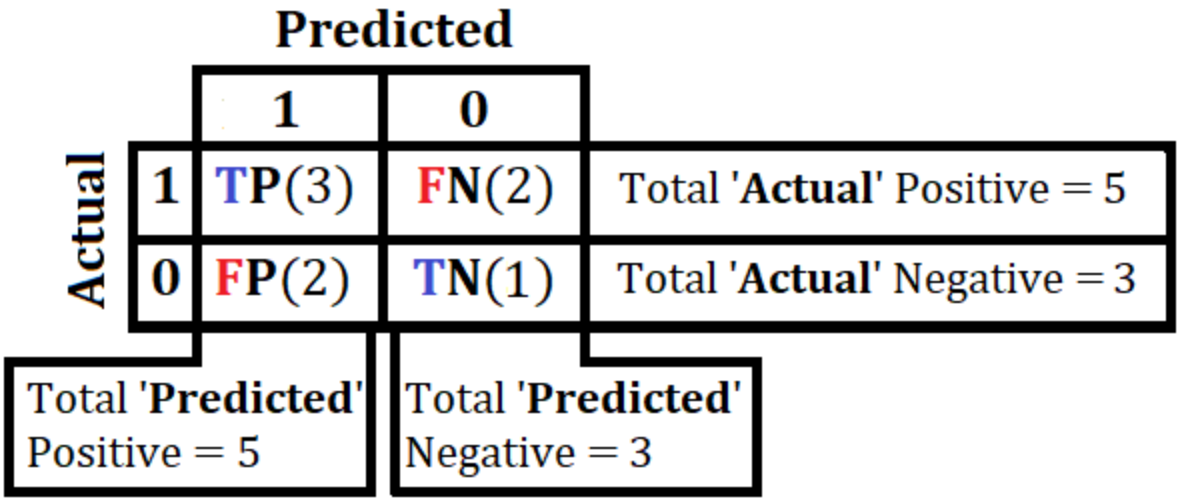


$$\textbf{Accuracy} = \frac{\text{Total Corrected Predictions(doesn't matter 0 or 1)}}{\text{N (total row numbers i.e. total prediction numbers)}}$$

$$\textbf{Recall} = \frac{\textbf{Actual} \text{ Positive (TP)}}{\text{Total } \textbf{Actual} \text{ Positive (TP + FN)}}$$

$$\textbf{Precision} = \frac{\textbf{Actual} \text{ Positive (TP)}}{\text{Total } \textbf{Predicted} \text{ Positive (TP + FP)}}$$

$$\textbf{F1 Score} = 2 * \frac{\text{(Recall * Precision)}}{\text{(Recall + Precision)}}$$

When to use what ?

---

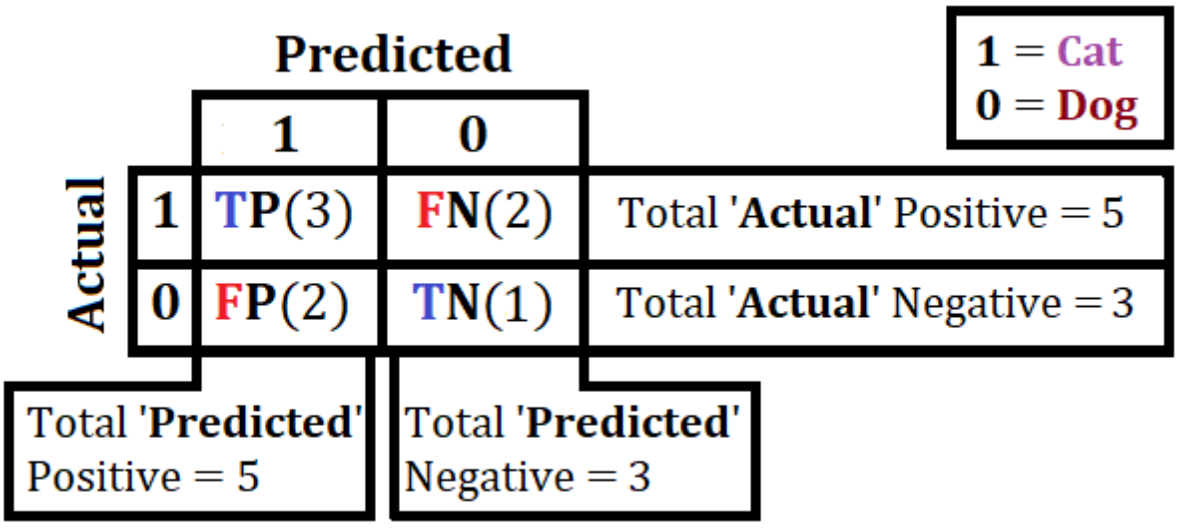
1) Accuracy is used when the dataset is balanced, or at least roughly.

2) For imbalanced dataset, recall and precision are used. Whether to use recall or precision, it depends what type of error(Type-1 Error = False Positive(FP), Type-1 Error = False Negative(FN)) is most important for our dataset and target.

3) F1 Score is used when we can't understand which type of error is important/dangerous for our purpose. In that case we need to look at both recall and precision and that's why precision was born so we don't have to look for both recall and precision separately.

Calculating Recall, Precision, F1 for each subclass :

In the first page, we've calculated recall, precision and F1 only for the first / positive row. What if we want to calculate both positive and negative separately?



For **Cat** : (1 / Positive)

$$\textbf{Recall} = \frac{\textbf{Actual Positive} \quad (TP)}{\text{Total } \textbf{Actual} \text{ Positive } (TP + FN)} = \frac{3}{3 + 2} = 0.6$$

$$\textbf{Precision} = \frac{\textbf{Actual Positive} \quad (TP)}{\text{Total } \textbf{Predicted} \text{ Positive } (TP + FP)} = \frac{3}{3 + 2} = 0.6$$

$$\textbf{F1 Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} = 2 * \frac{0.6 * 0.6}{0.6 + 0.6} = 0.6$$

For **Dog** : (0 / Negative)

$$\textbf{Recall} = \frac{\textbf{Actual Negative} \quad (TN)}{\text{Total } \textbf{Actual} \text{ Negative } (FP + TN)} = \frac{1}{2 + 1} = 0.3$$

$$\textbf{Precision} = \frac{\textbf{Actual Negative} \quad (TN)}{\text{Total } \textbf{Predicted} \text{ Negative } (FN + TN)} = \frac{1}{2 + 1} = 0.3$$

$$\textbf{F1 Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} = 2 * \frac{0.3 * 0.3}{0.3 + 0.3} = 0.3$$

```python
import numpy as np
from sklearn.metrics import accuracy_score, recall_score, precision_score,
f1_score

y =      np.array([1, 1, 1, 1, 1, 0, 0, 0])
y_pred = np.array([1, 1, 1, 0, 0, 0, 1, 1])

print(f"Accuracy =  {accuracy_score( y_true = y, y_pred = y_pred)}.")
# Accuracy is used for whole dataset. So can't use "average=None".
```

```
print(f"Recall    = {recall_score(   y_true = y, y_pred = y_pred,
average=None)}.")
print(f"Precision = {precision_score(y_true = y, y_pred = y_pred,
average=None)}.")
print(f"F1 Score  = {f1_score(       y_true = y, y_pred = y_pred,
average=None)}.")

# Output :
Accuracy =  0.5.
Recall    = [0.33333333 0.6       ].
Precision = [0.33333333 0.6       ].
F1 Score  = [0.33333333 0.6       ].
```

The first value is for 0(Dog), then 1(Cat).

How about calculating the **Confusion Matrix** ?

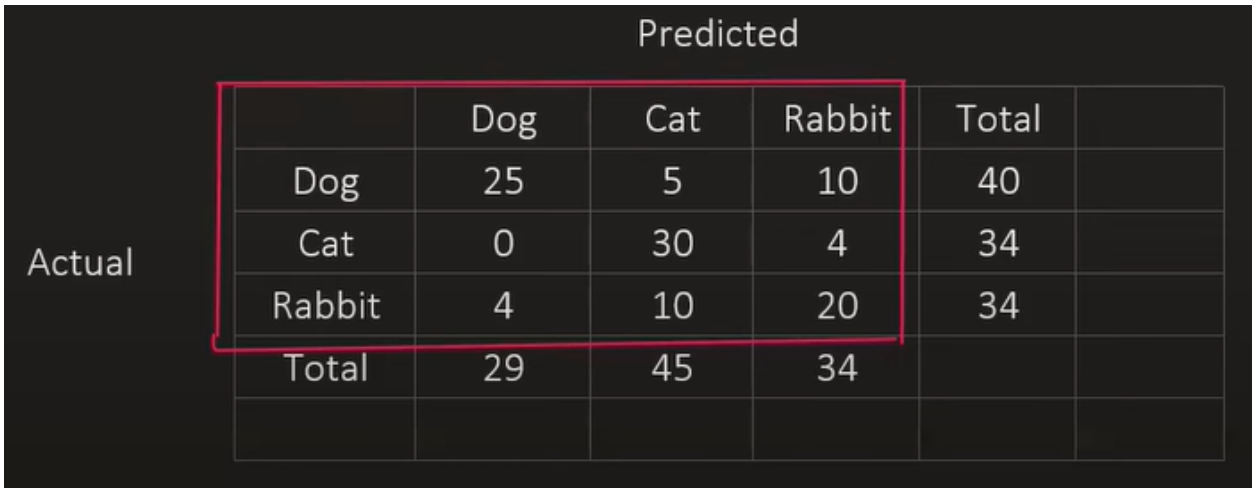
```
import numpy as np
import polars as pl
from sklearn.metrics import confusion_matrix

y =      np.array([1, 1, 1, 1, 1, 0, 0, 0])
y_pred = np.array([1, 1, 1, 0, 0, 0, 1, 1])

confu_mat = confusion_matrix(y_true = y, y_pred = y_pred) # returns 2D Array.
confusionMatrix = pl.DataFrame(data={'0' : confu_mat[:, 0],
                                     '1' : confu_mat[:, 1]}).with_row_index('')
print(confusionMatrix)

output = """
                                  ┌─────┬─────┐
                                  ¦ 0   ¦ 1   ¦
                              ╞═══╪═════╪═════╡
                              ¦ 0 ¦ 1   ¦ 2   ¦
                              ¦ 1 ¦ 2   ¦ 3   ¦
                              └───┴─────┴─────┘
"""
```

Multiclass Precision and Recall : (from Campusx)



Actual \ Predicted	Dog	Cat	Rabbit	Total
Dog	25	5	10	40
Cat	0	30	4	34
Rabbit	4	10	20	34
Total	29	45	34	

Recall :

For **Dog** :

$$= \frac{\textbf{Actual Dog}}{\text{Total } \textbf{Actual Dog}} = \frac{25}{40} = 0.625$$

For **Cat** :

$$= \frac{\textbf{Actual Cat}}{\text{Total } \textbf{Actual Cat}} = \frac{30}{34} = 0.88$$

For **Rabbit** :

$$= \frac{\textbf{Actual Rabbit}}{\text{Total } \textbf{Actual Rabbit}} = \frac{20}{34} = 0.58$$

Okay. we've calculated Recall separately for each class but how to calculate overall?

There are 2 ways : (R_{Dog} = 0.625, R_{Cat} = 0.88, R_{Rabbit} = 0.58)

1. Macro : (Just calculate the **average**)

$$\text{Macro} = \frac{R_{Dog} + R_{Cat} + R_{Rabbit}}{\text{Total class number}}$$

$$= \frac{0.625 + 0.88 + 0.58}{3}$$

$$= 0.695$$

2. Weighted :

$$\text{Weighted} = R_{Dog} * \text{Weight}_{Dog} + R_{Cat} * \text{Weight}_{Cat} + R_{Rabbit} * \text{Weight}_{Rabbit}$$

(Weight_{Dog} = Total Actual Dog / All Actual class
= 40 / (40 + 34 + 34)
= 40 / 108)

So, Weighted = 0.625 * (40/108) + 0.88 * (34/108) + 0.58 * (34/108)
= 0.691

Use Macro when the dataset is almost balanced, otherwise Weighted.

Similarly we can calculate for precision also.

```
import numpy as np
from sklearn.metrics import accuracy_score, recall_score, precision_score,
f1_score

y =      np.array([1, 1, 1, 1, 1, 0, 0, 0])
y_pred = np.array([1, 1, 1, 0, 0, 0, 1, 1])

print(f"Accuracy =  {accuracy_score( y_true = y, y_pred = y_pred)}.\n")
# Accuracy is used for whole dataset. So can't use "average=None".
```

```
#                    Recall, Precision, F1 Score directly.
print(f"Recall    = {recall_score(   y_true = y, y_pred = y_pred)}.")
print(f"Precision = {precision_score(y_true = y, y_pred = y_pred)}.")
print(f"F1 Score  = {f1_score(       y_true = y, y_pred = y_pred)}.\n")

#                    Recall, Precision, F1 Score by changing the parameter
'average'.
print(f"Recall    = {recall_score(   y_true = y, y_pred = y_pred, average =
'macro') : .2f}.")
print(f"Precision = {precision_score(y_true = y, y_pred = y_pred, average =
'weighted')}.")
print(f"F1 Score  = {f1_score(       y_true = y, y_pred = y_pred, average =
'weighted')}.")

# Output :
Accuracy =  0.5.

Recall    = 0.6.
Precision = 0.6.
F1 Score  = 0.6.

Recall    =  0.47.
Precision = 0.5.
F1 Score  = 0.5.
```