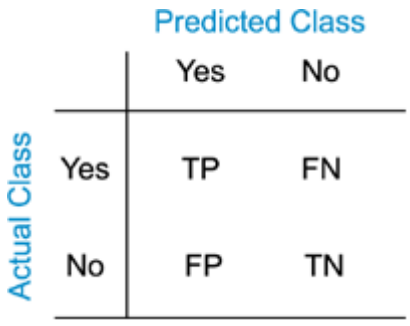
Confusion Matrix & Accuracy Metrics

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

- **TP** = true positives: number of examples predicted positive that are actually positive.
- **FP** = false positives: number of examples predicted positive that are actually negative.
- **TN** = true negatives: number of examples predicted negative that are actually negative.
- **FN** = false negatives: number of examples predicted negative that are actually positive.



- **True-Positive Rate** = TP / TP + FN
- **False-Positive Rate** = FP / FP + TN
- **True-Negative Rate** = TN / TN + FP
- **False-Negative Rate** = FN / FN + TP

Classifier Accuracy, or recognition rate: percentage of test set instances that are correctly classified

- Accuracy = (TP + TN)/All
- Error rate: 1 – accuracy, or Error rate = (FP + FN)/All

Class Imbalance Problem: One class may be rare, e.g. fraud, or HIV-positive

- Sensitivity: True Positive recognition rate = TP/P
- Specificity: True Negative recognition rate = TN/N

Other Classifier Evaluation Metrics

- **Precision**
  - % of instances that the classifier predicted as positive that are actually positive

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

- **Recall**
  - % of positive instances that the classifier predicted correctly as positive
  - a.k.a "Completeness"

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

- **Perfect score for both is 1.0, but there is often a trade-off between Precision and Recall**
- ***F* measure (F_1 or *F*-score)**
  - harmonic mean of precision and recall

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Sample Problem

1.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          71               71      %
Incorrectly Classified Instances        29               29      %
Kappa statistic                          0.3108
Mean absolute error                      0.3333
Root mean squared error                  0.4662
Relative absolute error                 69.9453 %
Root relative squared error             95.5466 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.967     0.692      0.686     0.967     0.803      0.709    0
                 0.308     0.033      0.857     0.308     0.453      0.708    1
Weighted Avg.    0.71      0.435      0.753     0.71      0.666      0.709

=== Confusion Matrix ===

  a  b   <-- classified as
 59  2 |  a = 0
 27 12 |  b = 1
```

2.

```
Number of Leaves  :     28
Size of the tree :  43
Time taken to build model: 0.18 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        1774               59.1333 %
Incorrectly Classified Instances      1226               40.8667 %
Kappa statistic                          0.1807
Mean absolute error                      0.4773
Root mean squared error                  0.4885
Relative absolute error                 95.4768 %
Root relative squared error             97.7122 %
Total Number of Instances             3000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC
Area  Class
                 0.662     0.481      0.587     0.662     0.622      0.616
1
                 0.519     0.338      0.597     0.519     0.555      0.616
0
Weighted Avg.    0.591     0.411      0.592     0.591     0.589      0.616
```

```
=== Confusion Matrix ===

    a    b   <-- classified as
 1009  516 |    a = 1
 710  765 |    b = 0
```

resources:
https://www.researchgate.net/post/How_do_I_calculate_the_false_alarm_rate_for_face_detection
https://www.ibm.com/developerworks/library/os-weka2/
https://machinelearningmastery.com/use-classification-machine-learning-algorithms-weka/