Performance measures of a binary classification

The **performance** of pattern recognition system can be very well characterized by the **recognition rates** considered for each class considering a **binary classification** problem.

Thus, for each class

- the example (instance, sample) belonging to the class is considered **positive**
- the example (instance, sample) not belonging to the class is considered **negative**.

Classification results

- **True positive *TP* (**correctly classified as positive; ***hit*****)** – an instance belonging to the class is correctly classified as belonging to the class;
- **True negative *TN* (**correctly classified as negative, ***correct rejection*****)** – an instance that does not belong to the class is correctly classified as not belonging to the class;

- **False pozitiv *FP* (**incorrectly classified as positive, ***false alarm*****)** – an instance that does not belong to the class is classified (incorrectly) as belonging to the class;
- **False negative *FN* (**incorrectly classified as negative, ***miss*****)** – an instance belonging to the class is classified (incorrectly) as not belonging to the class;

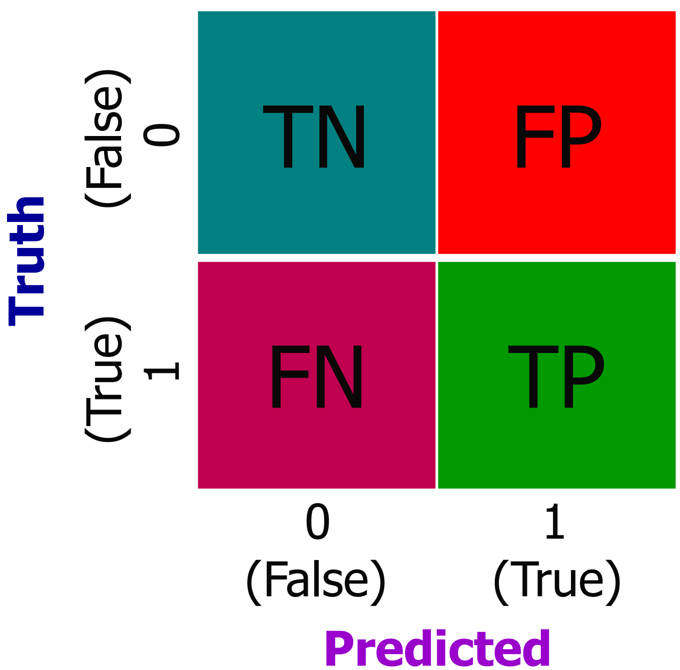
Confusion matrix

The confusion matrix (error matrix) provides a granular way to evaluate the results of a classification algorithm than just accuracy.
It does this by dividing the results into two categories that join together within the matrix:

- **Truth labels (targets) – actual class**
- **Predicted labels (outputs of the model) – predicted class**

Each row in a confusion matrix represents an actual class

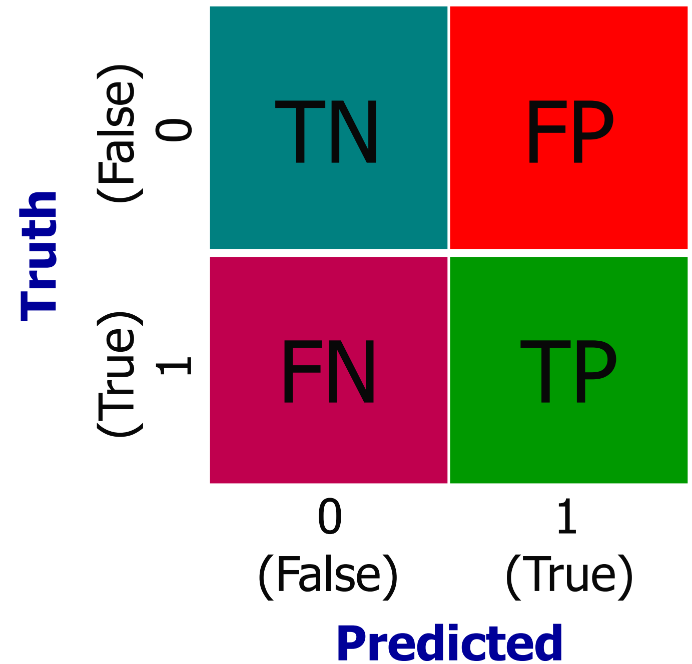
Each column represents a predicted class



True / False
from the prediction perspective

- **[1][1]** represents the values which are predicted to **be false and are false**. **TN**
- **[2][1]** represents the values which are predicted to **be false but are true**. **FN**
- **[1][2]** represents the values which are predicted to **be true but are false**. **FP**
- **[2][2]** represents the values which are predicted to **be true and are true**. **TP**

- **True negative *TN* (**correctly classified as negative, ***correct rejection*****)** – an instance that does not belong to the class is correctly classified as not belonging to the class;
- **True positive *TP* (**correctly classified as positive; ***hit*****)** – an instance belonging to the class is correctly classified as belonging to the class;
- **False pozitiv *FP* (**incorrectly classified as positive, ***false alarm*****)** – an instance that does not belong to the class is classified (incorrectly) as belonging to the class;
- **False negative *FN* (**incorrectly classified as negative, ***miss*****)** – an instance belonging to the class is classified (incorrectly) as not belonging to the class;



Recognition rates

- **True positive rate (*sensitivity; recall; hit rate*):**

$$\boldsymbol{TPR} = TP \,/\, P = TP \,/\, (TP + FN)$$

Correctly classified as positive out of all positive examples

- **True negative rate (*specificity; selectivity*):**

$$\boldsymbol{TNR} = TN \,/\, N = TN \,/\, (TN + FP)$$

Correctly classified as negative out of all negative examples

- **False positive rate (*fall-out*):**

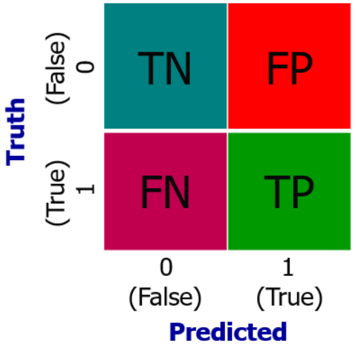
$$\boldsymbol{FPR} = FP \,/\, N = FP \,/\, (TN + FP)$$

Incorrectly classified as positive out of all negative examples

- **False negative rate (miss rate):**

$$\boldsymbol{FNR} = FN \,/\, P = FN \,/\, (TP + FN)$$

Incorrectly classified as negative out of all positive examples



$$TPR + FNR = 1$$

$$TNR + FPR = 1$$

The 3 pillars of binary classification

- **Precision (*positive predicted values*):**

$$PPV = TP \,/\, (TP + FP)$$

Correctly classified as positive out of all positive predictions

The success probability of making a correct positive class prediction

- **Accuracy:**

$$ACC = (TP + TN) \,/\, (P + N) = (TP + TN) \,/\, (TP + FN + TN + FP)$$

Success probability of detecting the positive and negative classes

(# correct predictions / # total predictions)

- **Recall *(TPR)*:**

$$Recall = TP \,/\, P = TP \,/\, (TP + FN)$$

How sensitive the model is towards identifying the positive class

$$f1-score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

f1-score is the harmonic mean of precision and recall (a combined metric)

f1-score gives a higher weight to low values.

We can only have a high *f1-score* if both precision and recall are high.

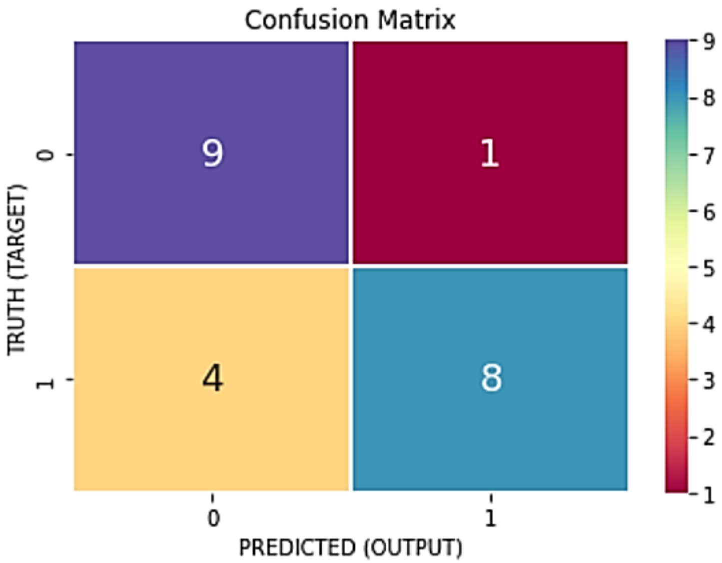
Illustration

>> truth: [1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0]
>> predicted: [0 1 1 0 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0]

`sklearn.metrics.classification_report(truth, predicted)`

	precision	recall	f1-score	support
0	0.69	0.90	0.78	10
1	0.89	0.67	0.76	12
accuracy			0.77	22
macro avg	0.79	0.78	0.77	22
weighted avg	0.80	0.77	0.77	22

`metrics.confusion_matrix(truth, predicted)`



macro average
averaging the unweighted mean per label

weighted average
averaging the support-weighted mean per label

Exercise

Determine:

TP
TN
FP
FN

TPR
TNR
FPR
FNR

Precision
Accuracy
Recall

f1-score

Confusion matrix for multiclass classification

```
truth:     [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0]
predicted: [2 1 1 2 2 2 2 0 2 2 2 2 1 2 1 0 2 1 0 1 1 1 2 1 1 1 2 0 0 0 0 2 0 0 0 0 0]
```

```
>>  Metrics classification report:
```

	precision	recall	f1-score	support
0	0.75	0.90	0.82	10
1	0.64	0.58	0.61	12
2	0.71	0.67	0.69	15
accuracy			0.70	37
macro avg	0.70	0.72	0.71	37
weighted avg	0.70	0.70	0.70	37

$$ACC = \frac{\#\ correct\ predictions}{\#\ examples}$$

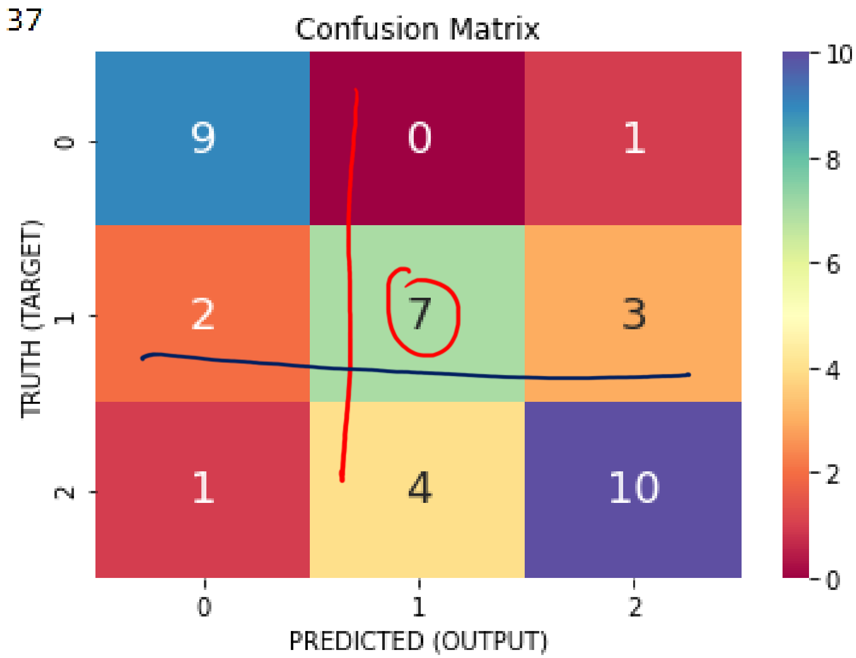
$$Acc = \frac{9+7+10}{10+12+15} = \frac{26}{37} = 0.703$$

precision in class „1"

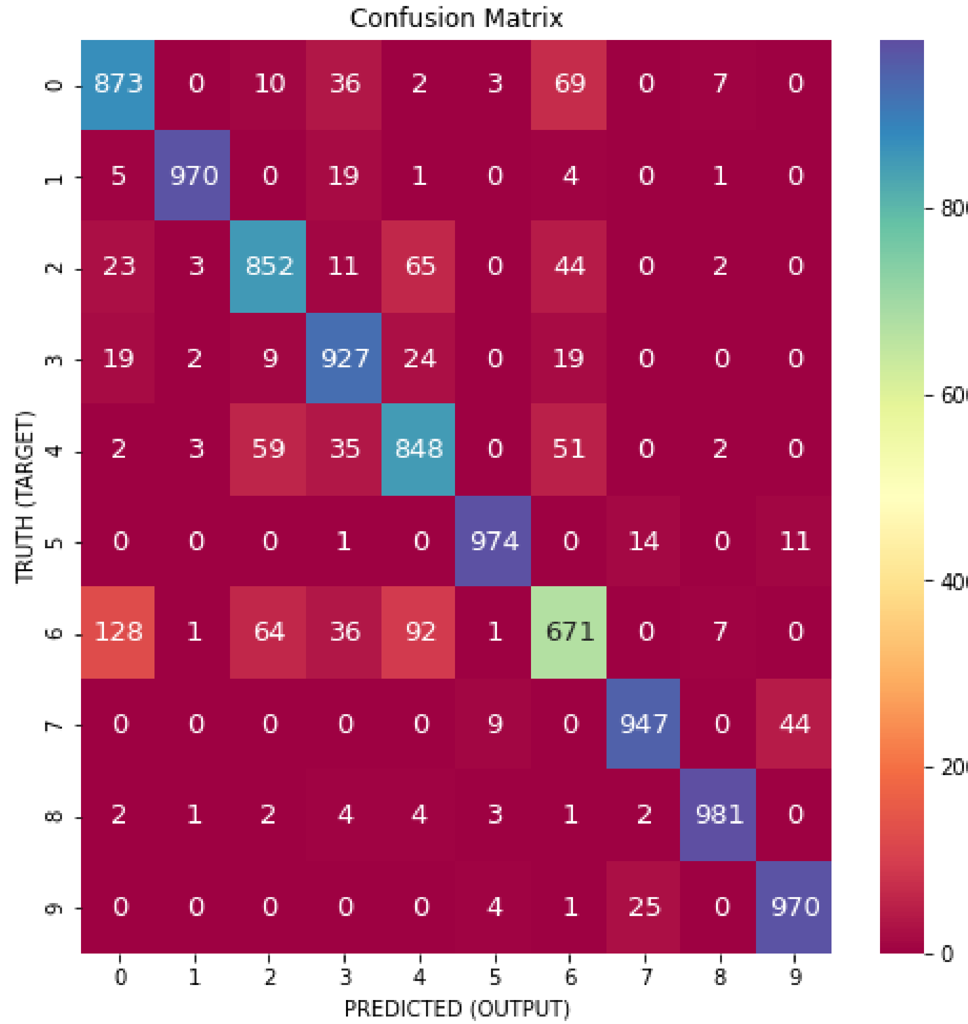
$$\frac{7}{0+7+4} = \frac{7}{11} = 0.64$$

recall „1"

$$\frac{7}{2+7+3} = \frac{7}{12} = 0.58$$



CNN implementation using TensorFlow2 to classify the MNIST FASION dataset




```
>>  Metrics classification report:

              precision    recall  f1-score   support

           0       0.83      0.87      0.85      1000
           1       0.99      0.97      0.98      1000
           2       0.86      0.85      0.85      1000
           3       0.87      0.93      0.90      1000
           4       0.82      0.85      0.83      1000
           5       0.98      0.97      0.98      1000
           6       0.78      0.67      0.72      1000
           7       0.96      0.95      0.95      1000
           8       0.98      0.98      0.98      1000
           9       0.95      0.97      0.96      1000

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000
```