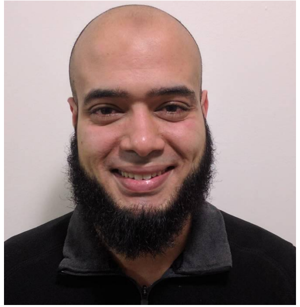
# Machine Learning

# Evaluation Metrics 2

**Mostafa S. Ibrahim**
*Teaching, Training and Coaching for more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*
*PhD* from Simon Fraser University - Canada
*Bachelor / MSc* from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



# Performance/Error Analysis

- By setting a specific threshold, we classify predictions as positive or negative
  - Varying thresholds yield different outcomes
- Comprehending your models stands as a crucial and challenging skill
- Transitioning between domains necessitates substantial ad-hoc effort
- Assume in a balanced dataset, the performance is 80%
- Your team aims for 97% accuracy
- How can you gain insights from that 80% accuracy?

# Performance/Error Analysis

- Within our dataset, 80% represent successful scenarios, while 20% depict failures
    - For each scenario, predictions can be true or false
- Your classifier says true (positive/1) or false (negative/0)
    - Hence, there are two possibilities
- The classifier's outcomes are either correct (true) or incorrect (false)
    - Again, there are two options
- Overall, this leads to four distinct scenarios!

# Performance/Error Analysis

- True Positives (TP): number of examples where your **model** correctly predicts **positive** and this is a **correct/true** prediction according to the ground truth
- True Negatives (TN): Models predicts negative and this is true
- False Positives (FP): Models predicts positive but this is wrong/false
    - Known as Type I errors
- False Negatives (FN): the model wrongly predicts negative
    - Known as Type II errors
- FP and FN refers to **incorrect** predictions according to the ground truth
- The **second** word (positive/negative) refers to the model’s prediction
- The **first** word (true/false) judges the model's accuracy

# Question!

- Compute four values (**tn, fp, fn, tp**) from the following results:
- y_pred = [0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]    From the model
- y_true = [0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]    From ground truth
- Remember:
    - True Negatives (TN): Model 0 and matches GT    (m=0, gt=0)
    - False Positives (FP): Model 0 but doesn't match GT    (m=0, gt=1)
    - True Positives (TP): Model 1 and matches GT    (m=1, gt=1)
    - False Negatives (FN): Model 1 but doesn't match GT    (m=1, gt=0)

**tn=2, fp=4, fn=5, tp=3**

# Confusion Matrix

- For a binary classifier, it is **2x2 matrix**/table that consists of the previous 4 numbers (or percentages)
  - TP, TN, FP, FN
- y_pred = [0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0] From the model
- y_true = [0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1] From ground truth
  - tn=2, fp=4, fn=5, tp=3

| | |
|---|---|
| tn=2 | fp=4 |
| fn=5 | tp=3 |

# Confusion Matrix

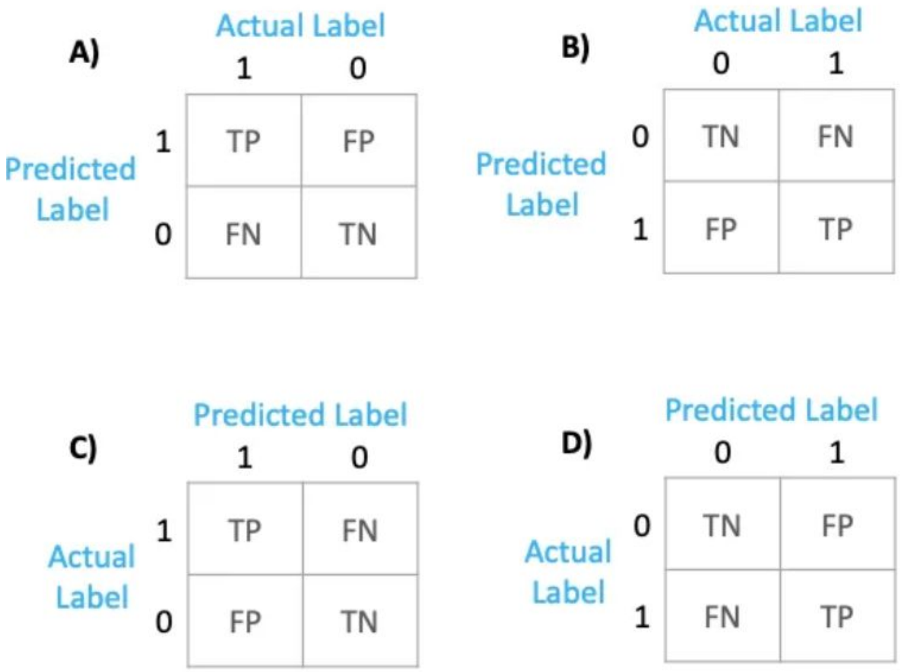
- In scikit, the row perspective represents the ground truth (actual) while column represents the model prediction

| | | **Predicted** | |
|---|---|---|---|
| | | Negative (**N**)<br>**-** | Positive (**P**)<br>**+** |
| **Actual** | Negative<br>**-** | True Negatives (T**N**) | False Positives (F**P**)<br>**Type I error** |
| | Positive<br>**+** | False Negatives (F**N**)<br>**Type II error** | True Positives (T**P**) |

Img src

# Which matrix?

- It's crucial to understand the order when you read/write a confusion matrix
- **Scikit** uses D's order
- Some lectures use order B
- I think A and C are rare




Img src

# Sklearn

```python
from sklearn.metrics import confusion_matrix

def cm():
    y_true = [0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1]     # ground
    y_pred = [0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]     # model

    conf = confusion_matrix(y_true, y_pred)
    print(conf)

    tn, fp, fn, tp = conf.ravel()   # table order

    print(f'tp={tp}, fn={fn}, tn={tn}, fp={fp}')

    '''

    [[2 4]
     [5 3]]
    tn=2, fp=4, fn=5, tp=3
```

# Informal Guidance on Relationships

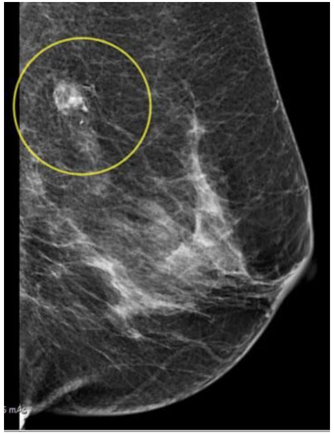
| | |
|---|---|
| tn | fp |
| fn | tp |

- There are two equations based on the ground truth (*inverse-like*):
    - (all gt negative examples) N = TN + FP (first row)
    - (all gt positive examples) P = TP + FN (second row)
- A higher threshold means the model is very restrictive on predicting positives
    - As a result, both TP and FP are *probably* reduced (FP and TP are positively related)
        - But with a strong model, TP could be high and FP could be low
    - If we label a few examples as positive, then we label more as negative!
        - Then FN is *probably* **inversely related** with FP / TP
- TN and FN: No direct relationship
    - If you are good at identifying negative examples (TN), you would likely reduce FN (inverse)
- TN and TP: No direct relationship (dependent on the model's performance)
- Still such thoughts mayn't hold strongly

# Question!

- Imagine our Breast Cancer Classifier is tested on three women
  - Assume has-cancer is the positive class
- 1: Sara has cancer. However, the model classifies her as not having cancer
  - Select from: FP, FN, TP, TN
- 2: Maryam is cancer-free. However, the model categorizes her as having cancer
  - Select from: FP, FN, TP, TN
- 3: Lila has cancer and the model accurately affirms this
  - Select from: FP, FN, TP, TN

# The cost of a mistake!



- When our classifier performs effectively (TP, TN), it's a satisfying outcome
- However, errors like FP and FN can pose challenges
- Assume our Breast Cancer Classifier is applied to two women
  - Assume positive means a cancer case
  - Maryam is cancer-free. However, the model identifies her as having cancer (FP)
  - Sara has cancer. Yet, the model classifies her as not having cancer (FN)
- Which mistake carries more weight and entails more substantial consequences?

# The cost of a mistake!

- When our classifier performs effectively (TP, TN), it's a satisfying outcome
- However, errors like FP and FN can pose challenges
- Let's consider our Spam Classifier applied to two emails, with "positive" indicating a spam email:
  - Email #1, from the government, is labeled as spam by the model (FP)
  - Email #2, from a hacker, is incorrectly marked as non-spam (FN)
- Which mistake holds greater significance and bears more substantial consequences?

# Question!

- We know the simple accuracy formula
- Build another formula for the accuracy from the confusion matrix

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

| | |
|---|---|
| tn=2 | fp=4 |
| fn=5 | tp=3 |

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

P = Number of positive examples = TP + FN
N = Number of negative examples = TN + FP

Notice Flip(TP) = FN

# More Metrics

- Several metrics were developed based on the confusion matrix
- They can provide more insights about the performance!
  - However, each metric has its own angle of evaluation

| | |
|---|---|
| tn | fp |
| fn | tp |

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."